

Universidad Politécnica de Madrid

On the Application of Meshfree
Methods to the Nonlinear
Dynamics of Multibody Systems

TESIS DOCTORAL

Daniel Iglesias Ibáñez
Ingeniero Industrial

2015

DEPARTAMENTO DE MECÁNICA DE MEDIOS CONTINUOS
Y TEORÍA DE ESTRUCTURAS
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
CAMINOS, CANALES Y PUERTOS

ON THE APPLICATION OF MESHFREE
METHODS TO THE NONLINEAR
DYNAMICS OF MULTIBODY SYSTEMS

TESIS DOCTORAL

AUTOR

DANIEL IGLESIAS IBÁÑEZ
INGENIERO INDUSTRIAL

DIRECTOR

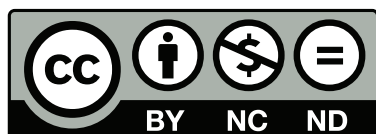
JUAN CARLOS GARCÍA ORDEN
DOCTOR INGENIERO AERONÁUTICO

The composition of the present text and pictures were made using \LaTeX , TikZ
Gnuplot, LibreOffice Calc, Inkscape, FreeCad, Salome, MknixPost & Ansys.

Oxford, UK, 2015.

Autor: Daniel Iglesias Ibáñez

Email: diglesias78@gmail.com





Tribunal nombrado por el Mgfc. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día 26 de Noviembre de 2015.

Presidente D.

Vocal D.

Vocal D.

Vocal D.

Secretario D.

Realizado el acto de defensa y lectura de la Tesis el día de de 2016 en la E.T.S. de Ingenieros de Caminos, Canales y Puertos de la U.P.M.

Calificación:

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO

*Talk is cheap.
Show me the code.*

—Linus Torvalds

All tools developed for this thesis are available through Github under GPL & LGPL free licenses:

LMX Algebra, solvers, and integrators library: <http://daniel-iglesias.github.io/lmx/>

MkniX Complete multibody simulation code: <http://daniel-iglesias.github.io/mknix/>

MkniXPost 3D graphical post-processor: <http://daniel-iglesias.github.io/mknixpost/>

Abstract

A new complete formulation is proposed for the simulation of nonlinear dynamic of multibody systems with thermo-mechanical behaviour. The approach is founded in three main pillars: total Lagrangian formulation, Bubnov-Galerkin discretization, and meshfree shape functions. Meshfree methods are characterized by the definition of a set of shape functions in overlapping domains, and a background grid for integration of the Galerkin discrete equations.

Two different types of shape functions have been chosen as representatives of interpolation (Radial Basis Functions), and approximation (Moving Least Squares) families. Their formulation has been adapted to use compatible parameters, and their lack of predefined connectivity is used to interconnect different domains seamlessly, allowing the use of non-conforming meshes.

A generalized formulation for constraints, joints, and contacts is proposed, which is valid for rigid and flexible solids, being the later discretized using either finite elements (FEM) or meshfree methods. The greatest advantage of this approach is that makes the domain completely independent of the external links and actions, allowing to even define them outside of the boundary. At the same time, the number of constraint equations needed for defining realistic joints is minimized.

Validation, examples, and benchmarks are provided for the proposed formulation, demonstrating that the approach is generic and extensible to further problems. Comparisons with FEM show a much lower error for the same number of nodes, both for mechanical and thermal analyses. The numerical efficiency is also better when coarse discretizations are used. A final demonstration to a real problem for handling massive structures inside of a fusion reactor is presented. It demonstrates that the application of meshfree methods is feasible and can provide an advantage towards the definition of nonlinear real-time simulation models.

Resumen

Esta tesis propone una completa formulación termo-mecánica para la simulación no-lineal de mecanismos flexibles basada en métodos libres de malla. El enfoque se basa en tres pilares principales: la formulación de Lagrangiano total para medios continuos, la discretización de Bubnov-Galerkin, y las funciones de forma libres de malla. Los métodos sin malla se caracterizan por la definición de un conjunto de funciones de forma en dominios solapados, junto con una malla de integración de las ecuaciones discretas de balance.

Dos tipos de funciones de forma se han seleccionado como representación de las familias interpolantes (Funciones de Base Radial) y aproximantes (Mínimos Cuadrados Móviles). Su formulación se ha adaptado haciendo sus parámetros compatibles, y su ausencia de conectividad predefinida se ha aprovechado para interconectar múltiples dominios de manera automática, permitiendo el uso de mallas de fondo no conformes.

Se propone una formulación generalizada de restricciones, juntas y contactos, válida para sólidos rígidos y flexibles, siendo estos últimos discretizados mediante elementos finitos (MEF) o libres de malla. La mayor ventaja de este enfoque reside en que independiza completamente el dominio con respecto de las uniones y acciones externas a cada sólido, permitiendo su definición incluso fuera del contorno. Al mismo tiempo, también se minimiza el número de ecuaciones de restricción necesarias para la definición de uniones realistas.

Las diversas validaciones, ejemplos y comparaciones detalladas muestran como el enfoque propuesto es genérico y extensible a un gran número de sistemas. En concreto, las comparaciones con el MEF indican una importante reducción del error para igual número de nodos, tanto en simulaciones mecánicas, como térmicas y termo-mecánicas acopladas. A igualdad de error, la eficiencia numérica de los métodos libres de malla es mayor que la del MEF cuanto más grosera es la discretización.

Finalmente, la formulación se aplica a un problema de diseño real sobre el mantenimiento de estructuras masivas en el interior de un reactor de fusión, demostrando su viabilidad en análisis de problemas reales, y a su vez mostrando su potencial para su uso en simulación en tiempo real de sistemas no-lineales.

Agradecimientos

A pesar de que la tesis está escrita en inglés, prefiero escribir estos agradecimientos en mi lengua materna, principalmente porque será la única manera de que mi padre lea lo que hemos conseguido. Sí, mamá, papá, esto lo empezásteis vosotros, y os lo debo todo. Ojalá estuviéramos los cuatro para celebrarlo, y no pasa un maldito día sin que piense en qué diría Esteban de las cosas que hacemos, juntos, separados, sonriéndonos, tirándonos los trastos a la cabeza, abrazándonos, o como sea. Es mi forma de que siga aquí, sobre todo para disfrutar de estos momentos buenos. Así que esto es por nosotros, por todo lo que hemos pasado. Gracias Familia.

Al haberme tomado la tesis como una especie de *hobby* en mis horas fuera de trabajo, he tenido la suerte de pasar por muchos centros y conocer muchísima gente que de una forma u otra han influido en su desarrollo. En la UPM arranqué junto con mi gran amigo Roberto, y allí coincidí a lo largo de los años con otros doctorandos. En especial Luis, Pablo y Miguel, he aprendido más de vosotros de lo que probablemente os podéis imaginar (a pesar de que alguno siga pensando que escribir *papers* en Word es una buena idea...). Tuve la suerte de encontrarme en un departamento lleno de profesores excepcionales, especialmente Juanjo, que la primera vez que vi cómo escudriñaba un código en C++, pensé: *yo tengo que aprender a hacer eso, pero ya*. Las explicaciones de Felipe acerca del FEM no lineal se me han quedado marcadas, también las clases de Ignacio, sobre todo una frase—probablemente distinta pero no en esencia—, *da igual cómo de complicado sea el problema que queréis simular, al final lo único que puede resolver el ordenador es $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$* . Y por supuesto mi tutor, Juan Carlos; como dijo Einstein y yo me permito adaptar a mi situación: *Hay dos cosas infinitas: el Universo y la paciencia de mi director de tesis. Y del Universo no estoy seguro*.

Pasé cinco años en Ciemat, y fue cuando realmente se me abrieron las puertas al mundo de la investigación en el campo de los aceleradores y de la fusión nuclear

(aunque se dice nucelar... nu-ce-lar). Aún recuerdo mi entrevista, y sé que es a Beatriz a quien tengo que agradecer que fuera el seleccionado. Espero que el trabajo que hice esos años estuviera a la altura de tus expectativas. Pero el Ciemat fue más que un inicio profesional, mucho más. No he encontrado grupo con tanta gente excepcional en ningún sitio, nunca. Sois muchos y no voy a nombraros a todos porque ni hace falta. Tanto hablando de trabajo, de actualidad o directamente de *bowling* y otras chorradas, habéis sido los mejores compañeros que probablemente tendré nunca. Cierto es que ha habido algunos destrozos por el camino, y seguís vetados para cualquier tipo de fiesta piscinera, pero eso es obviamente porque sois cojonudos.

Muchos colegas del CCFE van a tener que aprender *spanish* para ver que también han ayudado en la conclusión de este trabajo. Después de pasar dos años en *remote handling*, mis mejores recuerdos son para Keelan, que fue la primera persona que contraté y como sigue feliz en RACE, me alegro muchísimo de haber seleccionado a un Irlandés motivado en vez de a un militar estadounidense. Aunque no había ninguna duda en la elección, al final se ha confirmado como un acierto. Este último año he cambiado de tercio a Ingeniería Central y todavía estoy asimilando el buen hacer y la capacidad de la gente con que trabajo. Es un placer extremo compartir mi tiempo con este grupo de Profesionales, sí, en mayúsculas.

Escribí estas palabras en mis agradecimientos del DEA. Sigo pensando exactamente lo mismo que entonces, así que las reproduzco íntegramente aquí:

También deseo que conste mi gratitud a toda la comunidad que trabaja en favor del conocimiento libre, tanto científicos, educadores, programadores y todo aquel que aporte su trabajo al progreso de todos.

Empecé hablando de mi familia natural y quiero acabar dando mis más sinceras gracias a quienes sin serlo necesariamente de nacimiento, son también mi familia por todo lo que significan para mí. Primero, Mary, gracias a ti tengo la suerte de tener dos madres que me han enseñado todo, y que sobre todo me habéis establecido una escala de valores absolutamente impecable. Natalia, sigo confundiendo a la gente diciendo que somos primos porque compartimos apellido. Pero después de tanta broma, al final siento que podríamos ser perfectamente primos, o incluso si tuviera una hermana, no sería muy distinta a ti. Alicia, Bárbara, Beatriz, Diana, Emilia, Mabel y mis primos Paula y Carlos, me lleváis preguntando durante estos últimos meses (o años) cuándo iba a terminar la tesis. Bien, ya hay respuesta a la pregunta del millón, pero os aseguro que cada vez que preguntábais era un impulso enorme para seguir adelante.

Llego a mis mejores amigos, que son grandes no, enormes tampoco, lo siguiente. El hecho de estar alejado tanto tiempo de vosotros no ha cambiado ni un pelo lo cerca que me siento de vosotros. Juan Carlos y Lupo, os juro que voy a

ir a América a visitaros. Lo mismo te digo, Roberto—sí, vuelves a aparecer aquí—. Empezamos la tesis juntos y gran parte de ella te la debo a ti, porque fuiste la única persona—insistiéndome junto a Amandine—que me consiguió mantener la ilusión para seguir adelante durante muchísimo tiempo, incluso cuando dudaba si dejarla para centrarme en las actividades del CIEMAT. Juan, perrete bueno, mira que nos las has liado, maldito; tengo grabado en la memoria mi despedida guitarrera de Spain. Después de un año de ensayo, no quedo mal del todo, eh! De hecho, tengo ganas de hacer otra sesión de esas. Aparte de la música, me encanta que hayas encontrado un sitio donde puedas *frikear* a gusto, y en el que te valoren al menos por una mínima parte de lo que creo que vales. Y Marcos y Oriol, vosotros sois mis hermanos de adopción. Me da igual que nos conozcamos desde hace poco, algo me dice que nos vamos a tener que aguantar vaciles y coñas durante mucho tiempo. A Oriol ya se lo dije en su boda, que la vida me ha quitado un hermano, pero me lo ha compensado dándome otro. Realmente me ha dado dos, porque Marcos, respondiéndote a lo que escribiste en tu tesis, yo también te quiero, pero muchísimo; tenemos exactamente la misma sangre en las venas.

Finalmente, hay dos motivos por los que he podido terminar esta tesis. Primero, gracias al Ministerio por el cambio de ley, y sobre todo a la UPM por avisarme dos meses y medio antes de la fecha de deposición. No podría haber pasado tan buen rato sin vosotros. Para compensar, además de todo el apoyo de tanta gente, he tenido la suerte de contar con dos expertos internacionales que han revisado y recomendado el documento en un tiempo record: Rob Buckingham & Johannes Gerstmayr, *thank you so much!*

Pero ante todo, aquí en el exilio he encontrado a mi media naranja, quien pensando que estaba solo pendiente de sus problemas, a la vez se preocupaba por los dos. Carlota, esta es nuestra primera tesis; tú me has dado la fuerza para escribirla estos últimos meses. Ahora te toca a ti. Por suerte tienes más tiempo para organizarte y dedicarle el 100% del tiempo. Pero mientras investigas y la escribes, vamos a tener que acercarnos todavía un poquillo más. Eso ya sabes lo que es, porque ya te lo he pedido, y has dicho sí, y no hay nada que me haga más feliz.

Contents

1	Introduction	1
1.1	Motivation	1
1.1.1	DEMO remote handling systems	3
1.1.2	IFMIF accelerator beam facing components	6
1.2	Meshfree methods: State of the Art	8
1.3	Statement of the problem	10
1.4	Aims of the thesis	10
1.5	Scope, delimitation and methodology	10
1.6	Contents and structure of the document	12
2	Continuum Formulation of Flexible Bodies	13
2.1	Strain measures	15
2.2	Stress measures	15
2.3	Balance equations	15
2.3.1	Conservation of mass	16
2.3.2	Momentum balance	17
2.3.3	Balance of mechanical energy	19
2.3.4	Thermodynamics	19
2.4	Constitutive equations: hyperelasticity	22
2.5	Variational principles: weak formulation	23
2.5.1	Mechanical virtual work	23
2.5.2	Thermal virtual power	24
2.5.3	Modifications of the weak form	24
3	Meshfree Discretization of Flexible Bodies	27
3.1	Meshfree Functions	27

3.1.1	Properties of the meshfree shape functions	30
3.1.2	Moving Least Squares aproximation	30
3.1.3	Radial basis functions interpolation	33
3.2	Bubnov-Galerkin discretization of flexible bodies	37
3.2.1	Constrained Bubnov-Galerkin form	40
3.2.2	Transformation method	42
4	Flexible Multibody Systems Formulation	45
4.1	Balance equations of rigid bodies	46
4.2	Basic points coordinates	47
4.3	Formulation of rigid bodies	50
4.3.1	1-D bodies: Bars	51
4.3.2	2-D bodies: Plates	52
4.3.3	3-D bodies	52
4.4	Generalized joints formulation	53
4.4.1	Application of joints to flexible bodies	54
4.4.2	Mechanical constraint equations	55
4.4.3	Thermal constraint equations	64
4.4.4	Joints between rigid bodies	66
4.5	Generalized contact formulation	69
4.6	Proposed multibody formulation	72
5	Validation Cases	75
5.1	3-node triangle	75
5.1.1	Shape functions	75
5.1.2	Mechanical validation	80
5.2	4-node square	83
5.2.1	Discretization	83
5.2.2	Thermal validation	86
5.2.3	Mechanical validation	86
5.3	Prismatic beam	90
5.3.1	Beam discrete models	91
5.3.2	Results summary and discussion	97
6	Benchmarks	99
6.1	Mechanical benchmark	99
6.1.1	Error estimation	101
6.1.2	Results	102
6.2	Thermal benchmark	111
6.2.1	IFMIF LIPAc slits beam model	111

6.2.2	IFMIF LIPAc slits section analysis	114
6.3	Thermo-mechanical coupling benchmark	117
6.3.1	Overall models and simulation description	118
6.3.2	Initial heating	120
6.3.3	Cooling process	121
7	Application Examples and Simulation of Real Systems	125
7.1	Application examples	125
7.1.1	Generalized joints example	125
7.1.2	Hollowed double pendulums example	129
7.1.3	Generalized contact example	134
7.2	DEMO Remote Handling for blanket segments replacement	137
7.2.1	System description	137
7.2.2	Preliminary assessments and problem statement	138
7.2.3	Discrete models	141
7.2.4	COBS removal analyses	142
7.2.5	Computing efficiency	154
8	Conclusions and Outlook	157
8.1	Overview of main contributions	157
8.2	Summary of conclusions and research findings	158
8.3	Future work	160
	Bibliography	163
A	Contact forces with varying surface normal	171
A.1	2D case	171
B	Verification codes	173
B.1	Mechanical formulation	173
B.1.1	3-node triangle mechanical static	173
B.2	Thermal formulation	177
B.2.1	3-node triangle thermal static	177
B.2.2	Thermal dynamic	178

Introduction

1

1.1 Motivation

Flexibility in structural dynamics is usually discretized by means of the finite element method (FEM). In multibody dynamics, the computational cost of modelling a full mechanism considering all parts as continuum flexible bodies is so high that simplifications are needed for its analysis. In practice, mechanisms are composed of heterogeneous bodies with very different stiffness that justifies the use of models that consider the deformability of just the most flexible ones. Depending on the material, geometry, loading and the topology of the kinematic chain, alternative discretizations are used. A part can be considered as a fully rigid body, a body with small deformations with respect to a floating frame or a fully nonlinear deformable body. The bodies are linked through joints or contact constraints. We will present how these constraints can be applied between a body discretized using Galerkin meshfree methods and other parts of a multibody system, emphasizing its differences with the constraint formulation used in solid FEM discretization.

Thermal behavior is usually neglected when studying multibody systems, as the temperature variations are usually limited in magnitude and effect. In the case that the heat loads affect the global components of the system, or produce intense gradients, the thermal induced stress and deformations clearly influence the mechanics of the system. Also, the change in the configuration of the multibody system changes the thermal equilibrium and heat fluxes, resulting in a complex scenario for studying the temperature evolution inside of the different parts. Both cases can be better analyzed if the problem is posed using coupled field equations, and also can benefit from the use of meshfree methods for the spatial discretization of the continuum and joints.

Finite elements methods are not well suited to problems with extremely large

deformations [8], requiring complex and time consuming re-meshing or refinements during the simulation procedure. Another issue of these methods—due to the discontinuity of the derivatives of the shape functions—is that the representation of the stress and strain fields needs post processing techniques for recovering the continuity through elements. In meshfree methods, as there is no connectivity, large deformations do not reduce the accuracy of the shape functions' approximation. What is more, the influence domains overlap, assuring the continuity of the derivative fields. The higher precision in computing the stresses is specially meaningful in the boundary facets. On the other side, meshfree methods have a higher computational cost at equal number of DOFs.

In general, the higher cost of computing the shape functions in a meshfree method is worthwhile wherever the creation of a finite element mesh is a high time-consuming task or the problem requires frequent remeshings.

In multibody systems, this situation can appear when the flexible parts experience very large deformations. In compliant mechanisms, the substitution of discrete joints by localized ones (i.e. parts with areas that have more flexible behaviour) allows a dramatic reduction in production costs. In Micro-Electro-Mechanical Systems (MEMS), the flexibility of the parts can be the only way to achieve the desired kinematics of the mechanism.

In applied engineering problems it is usual to work with complex geometries. In these cases the creation of a proper mesh involves a very high human workload. Meshfree methods can give very accurate solutions using Delaunay meshes, which are guaranteed to exist in \mathbb{R}^2 . In \mathbb{R}^3 robust boundary conforming algorithms are available for the computation of tetrahedral meshes [67]. In FEM discretization, poorly shaped elements are usually the main source of numerical errors. In the methods presented in this paper, the mesh is only used for the numerical integration, which means that its geometrical characteristics do not affect the quality of the shape functions.

If low order triangular or tetrahedral FEM elements are used for nearly incompressible materials, numerical locking problems arise. This effect is usually avoided by means of higher order elements and mixed formulations, increasing the computational cost because of the higher number of degrees of freedom and a more complex formulation. Meshfree methods are not locking-free but due to the richness of their support function, it's so rare to find these kind of effects that the lack of this property needed to be proven [15].

All of these problems have been found in practical engineering designs and assessments of two different Fusion related projects. The common characteristic between them is the extremely challenging loads and environment conditions that the multiple-body systems need to withstand. From small devices subjected to very

localized heat loads, to large lifting robotic devices handling heavy radioactive components with very limited clearances. The analysis of their behaviour has demonstrated limitations on the usual approach using finite element discretization, which can be overcome using meshfree interpolation techniques.

1.1.1 DEMO remote handling systems

Fusion reactors are an excellent carbon-free alternative to help meet our future energy demands. Zero emissions, no long-lived radiation waste, abundant fuel supplies, and inherent safety are the most important advantages of magnetic confinement fusion reactors. On the downside, developing a fusion power plant represents an extremely challenging task from the physics, material technology, and engineering perspectives. The damage caused by highly energetic neutrons to the in-vessel components requires regular maintenance operations for their replacement.

The EU-DEMO (European Union DEMOnstration Power Plant) will be a nuclear fusion power plant that intended to build upon the scientific results of the ITER experimental nuclear fusion reactor. The objectives of DEMO lie between those of ITER and PROTO, which will be the prototype of future commercial power plants. Whereas ITER's goal is to produce 500 megawatts of fusion power (a source of heat) for at least 500 seconds, the goal of DEMO will be to produce at least four times that much fusion power on a much longer pulses, allowing to reach steady-state conditions of the tokamak. Moreover, while ITER's goal is to produce 10 times as much power as is required for breakeven, DEMO's goal is to produce 25 times as much power. DEMO's 2 to 3 gigawatts of thermal output will be on the scale of a modern electric power plant. Also, DEMO is notably intended to be the first fusion reactor to generate electrical power. Earlier experiments, such as ITER, merely dissipate the thermal power they produce into the atmosphere as steam and use stored tritium for fueling the tokamak, along with deuterium to generate the fusion plasma mix. Tritium is a very rare and expensive isotope, which shall be generated using the fusion neutrons for sustaining the reaction, dispensing with the fission reactor currently used for this purpose. DEMO will close both thermodynamic and fueling loops, demonstrating the overall feasibility of all the systems required for running a Fusion power plant.

To achieve its goals, DEMO must have linear dimensions about 15% larger than ITER and a plasma density about 30% greater than ITER. The European Roadmap to Fusion Energy [17] states the goal for DEMO of providing fusion energy to the grid by 2045. Subsequent commercial fusion reactors could be built for nearly a quarter of the cost of DEMO if things go according to plan.

While fusion reactors like ITER and DEMO will produce neither transuranic nor fission product wastes, which together make up the bulk of the nuclear wastes pro-

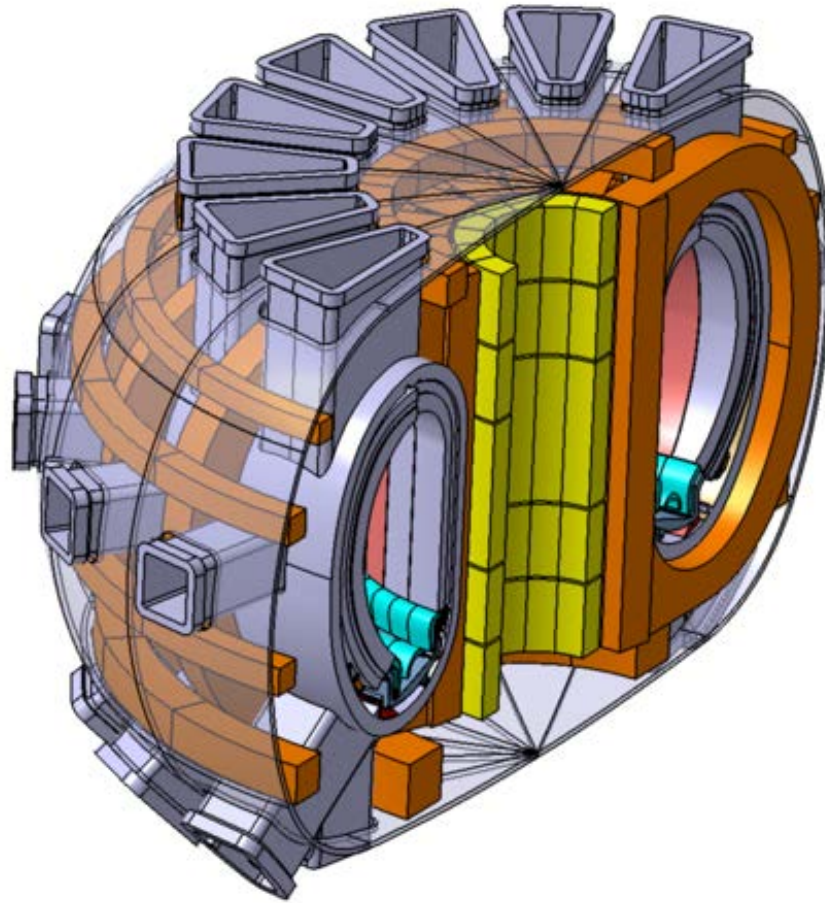


Figure 1.1: Sectioned baseline model of the European DEMO tokamak.

duced by fission reactors, the in-vessel components of the ITER and DEMO reactors will become radioactive due to neutrons impinging upon them. Plasma facing low activation materials, such as Eurofer, are in development so that wastes produced in this way will have much shorter half lives than the waste from fission reactors. The aim is to produce radioactive waste which remain harmful for less than one century. Qualification of these materials is the prime purpose of the International Fusion Materials Irradiation Facility.

During the shutdown of the reactor, the activated materials decay emitting gamma radiation. This ionizing field, along with the risk of contamination from activated dust, makes the use of remote handling equipment for all the in-vessel maintenance operations mandatory. Even for these mechatronic devices, the extremely high radiation fields in the range of kGy/h represents a huge constraint on their design [4]. The survival of their components is usually estimated by a simplistic approach. The maximum level of absorbed dose, along with the intervention time, gives an upper bound to the total damage. This has been sufficient for present experimental



Figure 1.2: Active blanket segment handling with remote handling equipment in yellow.

reactors, such as JET, as the radiation levels are orders of magnitude below those expected for future power plants. What is more, heavily radiated assemblies need to cope also with the thermal effects arising from the surface and volumetric heat generation.

Semi-automated remote handling systems will be in charge of removing the in-vessel components (IVC) once their service life expires [16]. Breeding blanket segments correspond to the most massive and challenging IVC to be handled, being around 10 m tall, weighting up to 80 t, and having only 20 mm gaps between them—which in fact is less than the expected structural deformation under their own weight. Their replacement with new segments will be a critical operation. Old segments will be highly active with a decay heat of several kW each, rising their bulk temperature over 100°C. The handling system will need to cope also with permanent deformations and unexpected changes in the structural behaviour [33]. Any failure will increase the downtime of the reactor, resulting in a very important loss of revenue. But, most importantly, all safety cases shall be studied to ensure that any scenario does not jeopardize the future operation of the plant. Credible demonstrations of these cases will be required by the Regulator for licensing the plant prior to the operations, which will involve analysing complex dynamic, nonlinear, thermo-mechanical, and radiation behaviours of the mechatronic systems while handling those massive components.

1.1.2 IFMIF accelerator beam facing components

The International Fusion Materials Irradiation Facility, also known as IFMIF, is a projected materials test facility in which candidate materials for the use in an energy producing fusion reactor can be fully qualified. IFMIF is an accelerator-based neutron source that produces, using deuterium-lithium nuclear reactions, a large neutron flux with a spectrum similar to that expected at the first wall of a fusion reactor. The IFMIF project was started in 1994 as an international scientific research program, carried out by Japan, the European Union, the United States, and Russia, and managed by the International Energy Agency. Since 2007, it has been pursued by the Japanese Government and EURATOM under the Broader Approach Agreement in the field of fusion energy research, through the IFMIF/EVEDA project, which conducts engineering validation and engineering design activities for IFMIF.

The IFMIF plant consists of five major systems [41]: 1) Accelerator Facility, 2) Li Target Facility, 3) Test Facility, 4) Post-Irradiation Examination (PIE) Facility, and 5) Conventional Facility. The whole plant must comply with international nuclear facility regulations. The energy of the beam (40 MeV) and the current of the parallel accelerators (2×125 mA) have been tuned to maximize the neutron flux ($10^{18} m^{-2} s^{-1}$) while creating irradiation conditions comparable to those in the first wall of a fusion reactor. Damage rates > 20 dpa per year of operation can be reached in a volume of 0.5 l of its High Flux Test Module that can accommodate around 1000 small specimens. The small specimen testing techniques developed aim at full mechanical characterization (fatigue, fracture toughness, crack growth rate, creep and tensile stress) of candidate materials, and allow, besides a scientific understanding of fusion neutron induced degradation phenomena, the creation of the major elements of a fusion materials database suited for designing, licensing and reliably operating future fusion reactors. The main expected contributions of IFMIF to the nuclear fusion community are to:

1. provide data for the engineering design for DEMO,
2. provide information to define performance limits of materials,
3. contribute to the completion and validation of existing databases,
4. contribute to the selection or optimization of different alternative fusion materials,
5. validate the fundamental understanding of the radiation response of materials including benchmarking of irradiation effects modelling at length-scales and time-scales relevant for engineering application, and

6. tests blanket concept and functional materials prior to or complementary to ITER test blanket module testing.

The IFMIF/EVEDA LIPAc [28] will be a 9 MeV, 125 mA cw deuteron prototype accelerator, identical to the low energy section of one of the 40 MeV IFMIF accelerators, which will be tested to verify the validity of the design before launching the IFMIF construction. It includes an ion source, a Radiofrequency Quadrupole cavity and the first module of a superconducting linac based on half wave resonator cavities. The accelerator is expected to generate the highest intensity continuous beam ever, producing an extremely high thermal load in all the elements intercepting the ions.

Independently of the final purpose of each device, if its working conditions imply stopping a non-negligible amount of particles, the associated thermal solicitation greatly determines the design constraints.

Different scrapers are located in the transport lines and a beam dump is required to stop the beam exiting the accelerator during commissioning and accelerator tests, so all these elements must operate in pulsed and CW modes.

A beam dump designed for a continuous maximum power of 1.125 MW will be used to stop the beam at the accelerator exit [30]. The conceptual design for the IFMIF and LIPAc accelerator beam dump is based on a conical beam stop made of OFE copper. The cooling system uses an axial high velocity flow of water pressurized up to $3.4 \cdot 10^5$ Pa to avoid boiling [59].

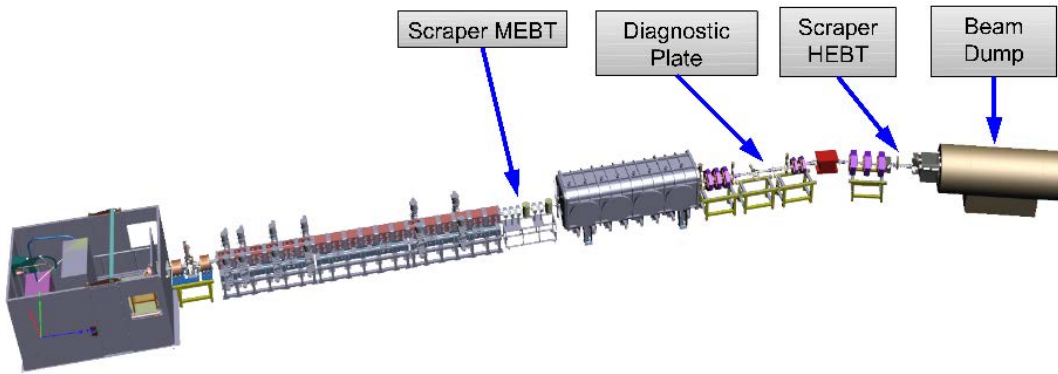


Figure 1.3: Scheme of the IFMIF/EVEDA accelerator, showing location of particle stopping devices.

Stopping beam particles on a safe and reliable way shall be the most important objective of the design, including radiation shielding [58]. The working conditions have to be simulated as best as reasonably possible, including normal, off-normal and accident situations.

The basic load that the material has to handle is the heat generation that comes from the ionization of the beam particles [10]. The usual way to describe the beam

is model it as a Gaussian distribution. The different acceleration stages of IFMIF LIPAc prototype produce a beam that is more squared so the information of the particle simulations must be used for the design.

The particles impinge upon the solid bodies of the beam facing devices, producing a thermal load as they lose their kinetic energy by an ionization process [31]. This load is usually modelled by a surface flux, as the ionization depth is in the order of microns. This approximation has proved to be too conservative in the design of pulsed beam diagnostics and dumps. An alternative approach to an extremely dense FEM mesh is pursued using meshfree methods, which shall be extendable to the study of the complex multi-body engineered systems.

1.2 Meshfree methods: State of the Art

There has been a growing interest on meshfree methods as an alternative to the widely used finite element methods [45] during the last decade. Their approach of mesh independent discretization by means of locally supported shape functions makes this type of methods well suited for those problems where the domain has a complex geometry or experiences very large deformations.

There is a general agreement to establish the origin of these methods in 1977, with the publication of the SPH method [53], explained in more detailed in [55]. The original application of the SPH was focused in astrophysics, but it has been extended later to other fields, as fluid and solid mechanics, and even used to couple fluid dynamics with multibody systems [65]

Another two development paths of meshfree methods where the generalization of the finite differences method and the particle in cell method. One of the first contributions to the first one is due to Perrone and Kao [60]. A remarkable contribution was the alternative technique proposed by Liszka and Orkisz [46] due to its robustness.

Until the 1990s, the development of the meshfree methods was limited, but during the last 23 years there has been a lot of activity in the development of new methods. One of these early methods proposed in that decade was the Diffuse Element Method (DEM) [56], based on a moving least square (MLS) approximation [43] and a discrete Bubnov-Galerkin form of continuum equations.

Using a similar focus but with different weight functions, Belytschko and co-workers develop the Element Free Galerkin (EFG) method [9]. These methods were first applied to fracture mechanics [62], adding contacts [6], and even using ALE formulations [61] but they have also been applied to general solid mechanics [63], and shell structures [49]. Even a test of using FEM compatible sliding joint was presented [37] modifying the EFG approximation functions to regain delta property at the ends of the element.

Simultaneously, W. K. Liu and co-workers presented the Reproducing Kernel Particle Method (RKPM) [50] in both continuous and discrete forms. It uses approximations in integral form in a similar way as SPH does, but enriched with the functions to be reproduced. It was applied initially to fluid mechanics and extended lately to the strong and weak formulation of solid mechanics, even under large deformations [12]. J.-S. Chen and co-workers developed a stabilized conforming nodal integration (SCNI) scheme for Galerkin meshfree methods, which allows the use of collocation techniques without the risk of arising instabilities in solid mechanics [72], and in Reisner-Mindling plates [71]. Extended meshfree methods use a combination of particular and homogeneous solutions to calculate the total solution, requiring much less computational effort [14]. Double grid approaches are presented for study of effects in meso-scale modelling of grain growth [13].

Multiquadrics [39] and other radial basis functions (RBFs) [40] have been applied to develop explicit collocation solvers for elliptic problems [38]. With a similar scheme as the EFG but using RBFs and with a possible enrichment with polynomials, the so-called Radial Point Interpolation Method (RPIM) is published by G. R. Liu and co-workers [47]. It has been applied to the linear formulation of solids, beams and plates. True meshless implementation of those shape functions was explored in [51], and further extended in [48]. As an alternative, using the method of fundamental solutions led to a boundary-type meshless method [44]. Local Petrov-Galerkin forms are proposed for elastic dynamic problems [52], and contact elastomeric components [29].

Natural neighbours [69] use Voronoi tessellation of sets of nodes. Elastoplastic 2D applications using local delaunay forms are described in [27].

Finally, the polygonal based interpolation by maximum entropy (max-ent) principle was introduced by N. Sukumar [68] and generalized for local schemes by [3]. The shape functions minimize the width of the support domain, interpolate affine functions exactly, and have a weak Kronecker-delta property at the boundary. The initial first order method has been extended to higher-order consistent interpolants using the Boor's algorithm [24]

Many other methods have been developed (MLPG, Hp-Clouds, Finite Spheres, etc.) but are not referenced and detailed here because they significantly differ from than the method proposed in this article, while providing no clear advantages in the context of multibody applications.

A detailed description of the origin and evolution of the meshfree methods can be found in the book [45], the review article [8], and other articles in the same monographic volume. Updated review of the different methods, including Matlab codes, is presented in [57], and the evolution of the shape function construction procedures can be found in [70].

1.3 Statement of the problem

Meshfree Galerkin methods have been developed in the last 20 years, and applied to the simulation of complex mechanical problems involving large strains of structures, crack propagation or high velocity impact dynamics. At the present time, the application of these methods to multibody dynamics has not been made despite their potential great advantage in some situations over standard finite element techniques. In particular, the following issues are aimed to be addressed:

- Increase the accuracy of thermo-mechanical simulations including large displacements and deformations.
- Simplify and generalize the joint definition and formulation, in a way that is independent on how the solid bodies are modelled.
- Improve the modelling time, as it usually takes around 80-90% of the total time spent by an analyst.
- Minimize the risk of mesh locking when using triangular or tetrahedral meshes.
- Reduce the computational time required to simulate a full nonlinear multibody system, aiming to provide soft real-time capabilities.

1.4 Aims of the thesis

The main objective of this research is to establish the suitability of meshfree methods for the space discretization of solid multibody systems using continuum mechanics formulations. In particular, the research aims to identify potential advantages of such discretizations, based in the items described in the previous section. The methodology shall be capable of enhancing existing state-of-the-art nonlinear simulation codes and algorithms, and its overall applicability will be demonstrated by the simulation of challenging multibody engineering problems.

1.5 Scope, delimitation and methodology

The project is centred in the application of 2D and 3D geometric nonlinear formulation embedded in a multibody framework. The proposed approach shall allow the implementation of flexible solid bodies with as minor changes as possible in a multibody code.

The types of problems of interest are dominated by thermo-mechanical effects, and have in common that the current state-of-the-art approach for solving them is limited, either by numerical stability, computational capacity, or accuracy of existing FEM based simulations.

No mechanical to thermal coupling effects are considered—apart from displacements and deformations—, so the following effects are neglected:

- Internal heat generation due to plastic deformations or viscous effects
- Heat generation due to friction

On the other side, thermal to mechanical coupling effects are considered through the next mechanism:

- Thermal expansion
- Temperature-dependent mechanical properties
- Heat transfer between contacting bodies

The flexibility is formulated by a Galerkin weak form. Among all the meshfree discretization methods, radial basis shape functions have been identified as the best ones for this kind of approach. The formulation is suited for both 2D and 3D problems, including static and dynamic analysis.

The proposed method uses a total Lagrangian formulation, such that shape functions are computed only at the beginning of the simulation. As exposed in following sections, this approach reduces significantly the number of operations, because the computation of the shape functions involves the resolution of a linear system for each Gauss point.

Among all parametrizations available to model solid bodies, generalized Cartesian natural coordinates is preferred when dealing with solid flexible bodies because their configuration is described by the coordinates of their nodes. This choice implies some extra nodes and constraints when modelling rigid bodies (e.g. a 3D rigid body needs four nodes and six constraints) but the cost is negligible compared to the DOFs that a flexible solid body usually has. This approach has numerous advantages, even when used only in rigid multibody systems [20], and also leads to a constant mass matrix for the whole system.

For the evaluation of the efficiency and complexity of the formulation presented, a multibody code will be developed from scratch. It will allow to run flexible multibody systems using both FEM and meshfree approximations. The requirements for its design are motivated for the need of sharing as much functions as possible between both approaches, in order to make a fair comparison between existing FEM techniques and the proposed meshfree methods applications. Its requirements and functionality are inspired in the SNAW code [26], which was written for the application of the RKPM method in the field of structural mechanics. It also has some similarities from the user's perspective to the multibody script language HOTINT [23].

1.6 Contents and structure of the document

The document is divided in two main parts. The formulation of the problem is described through chapters 2 to 4 while the numerical implementation, benchmarking, applications are covered on chapters 5 to 7. Each chapter can be briefly described as follows:

- Chapters 2 and 3 present the framework definition, corresponding to the adaptation of existing theories in nonlinear continuum mechanics and meshfree approximations, respectively. Work is also presented in chapter 3, defining a set of parameters valid for the definition of two different meshfree methods.
- Chapter 4 proposes a complete nonlinear multibody formulation including rigid bodies and generalized constraints, joints, and contacts.
- Chapter 5 Demonstrates the validity of the formulation.
- Chapter 6 Tests its performance compared to regular FEM.
- Chapter 7 Presents a set of examples that take advantage of the meshfree formulations, and also an application to the remote handling of the DEMO fusion reactor.

Continuum Formulation of Flexible Bodies

2

For the simulation of the flexibility of solids, the continuum mechanics theory defines the deformation as a change in the configuration of the body. The study of the effects that appear when a body changes from an undeformed (reference) configuration to a deformed (spatial) configuration can be done in both descriptions of the body.

Let \mathcal{B}_0 be the reference configuration (i.e. $t = 0$) and \mathcal{B}_t its configuration in time $t > 0$, we assume that there is a mapping:

$$\varphi : \mathcal{B}_0 \times [0, t) \rightarrow \mathcal{B}_t \in \mathbb{R}^3$$

that describes the *movement* of each particle of the body. This *motion* is described by:

$$\mathbf{x} = \varphi(\mathbf{X}, t) \quad (2.1)$$

where \mathbf{x} is the position of a point with material coordinate \mathbf{X} at time t .

The kinematic state of any point can be defined by its position and velocity. The difference between the deformed and the original configurations is described by the displacement. It can be expressed in the deformed system \mathbf{u} or in the reference one \mathbf{U} . Note that although it has two descriptions, it represents a unique motion:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{x} - \mathbf{X}(\mathbf{x}, t) \quad (2.2)$$

$$\mathbf{U}(\mathbf{X}, t) = \mathbf{x}(\mathbf{X}, t) - \mathbf{X} \quad (2.3)$$

$$\mathbf{U}(\mathbf{X}, t) = \mathbf{U}(\varphi^{-1}(\mathbf{x}, t), t) = \mathbf{u}(\mathbf{x}, t) \quad (2.4)$$

The velocity, $\mathbf{V}(\mathbf{X}, t)$, is the *material time derivative* of the displacement. This derivative is calculated with respect to the original configuration, \mathbf{X} and is also

called *total derivative* and noted as $D(\cdot)/Dt$ when indicial notation is used. As the time configuration \mathbf{x} is related to the initial configuration by means of the motion, the velocity can be expressed using the displacement or the position:

$$\mathbf{V}(\mathbf{X}, t) = \dot{\mathbf{U}} = \frac{\partial \mathbf{U}(\mathbf{X}, t)}{\partial t} = \frac{\partial \varphi(\mathbf{X}, t)}{\partial t} \quad (2.5)$$

Even though the velocity field is in essence a spatial vector as its existence is impossible without deformation, it is defined as a material variable. A spatial description of the velocity can be calculated using an inverse deformation:

$$\mathbf{v}(\mathbf{x}, t) = \dot{\mathbf{u}} = \frac{\partial \mathbf{x}}{\partial t} = \mathbf{V}(\varphi^{-1}(\mathbf{x}, t), t) \quad (2.6)$$

Finally, the acceleration, $\mathbf{A}(\mathbf{X}, t)$, is a derived quantity which describes the rate of change of the velocity:

$$\mathbf{A}(\mathbf{X}, t) = \dot{\mathbf{V}} = \frac{\partial \mathbf{V}(\mathbf{X}, t)}{\partial t} = \frac{\partial^2 \mathbf{u}(\mathbf{X}, t)}{\partial t^2} \quad (2.7)$$

And the relation between them is established in the deformed configuration:

$$\mathbf{a} = \dot{\mathbf{v}} = \dot{\mathbf{u}} \quad (2.8)$$

The evolution of the deformation is usually described by a *Lagrangian formulation*, which “follows” the displacement of each point in a fixed framework. If this formulation is described in the reference configuration, we will talk about a *material description* or *total Lagrangian formulation*. On the other hand, if the formulation is described in the deformed configuration, then a *spatial description* is used and the formulation is called *updated Lagrangian*.

An Eulerian description of a spatial variable can be derived also from its Lagrangian definition using the material derivative. As the former depends indirectly on the material coordinates, the chain rule can be used. Let the second order Cauchy stress tensor, $\boldsymbol{\sigma}(\mathbf{x}, t) = \sigma_{ij}$, be used as an example:

$$\frac{D\sigma_{ij}}{Dt} = \frac{\partial \sigma_{ij}}{\partial t} + \frac{\partial \sigma_{ij}}{\partial x_k} \frac{\partial x_k}{\partial t} \quad (2.9)$$

$$\dot{\boldsymbol{\sigma}} = \frac{\partial \boldsymbol{\sigma}}{\partial t} + \mathbf{v} \cdot \nabla \boldsymbol{\sigma} \quad (2.10)$$

The derivation leads to a sum of two terms, the *spatial derivative* and the *convective term*, respectively.

2.1 Strain measures

The *deformation gradient* second order tensor, $F(\mathbf{X}, t)$, is defined as the material derivative of the movement. It represents the relationship between the material coordinates \mathbf{X} and the spatial coordinates, \mathbf{x} :

$$\mathbf{F}(\mathbf{X}, t) \stackrel{\text{def}}{=} \frac{\partial \varphi(\mathbf{X}, t)}{\partial \mathbf{X}} \equiv \frac{\partial \mathbf{x}}{\partial \mathbf{X}}$$

Other useful measures can be defined using the deformation gradient:

- *Right Cauchy-Green deformation*: $\mathbf{C} \stackrel{\text{def}}{=} \mathbf{F}^T \mathbf{F}$
- *Left Cauchy-Green deformation*: $\mathbf{b} \stackrel{\text{def}}{=} \mathbf{F} \mathbf{F}^T$
- *Green-Lagrange strain*: $\mathbf{E} \stackrel{\text{def}}{=} \frac{1}{2}(\mathbf{C} - \mathbf{1})$
- *Almansi-Euler strain*: $\mathbf{e} \stackrel{\text{def}}{=} \frac{1}{2}(\mathbf{1} - \mathbf{b}^{-1})$
- *Euler linear strain*: $\boldsymbol{\varepsilon} \stackrel{\text{def}}{=} \frac{1}{2} \left[\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \right)^T \right]$

2.2 Stress measures

The Cauchy postulate establishes that the traction, \mathbf{t} , depends not only on the point in study, \mathbf{x} , but also on its associated surface. Using the normal of the surface, \mathbf{n} , the Cauchy second order stress tensor is defined as the one that verifies the following condition:

$$\mathbf{t}(\mathbf{x}, \mathbf{n}) = \boldsymbol{\sigma}(\mathbf{x}) \mathbf{n} \quad (2.11)$$

Using the Cauchy stress and the deformation gradient, the following stress measures are defined:

- *Piola-Kirchoff first tensor*: $\mathbf{P} \stackrel{\text{def}}{=} \det(\mathbf{F}) \boldsymbol{\sigma} \mathbf{F}^{-T}$
- *Piola-Kirchoff second tensor*: $\mathbf{S} \stackrel{\text{def}}{=} \mathbf{F}^{-1} \mathbf{P} = \det(\mathbf{F}) \mathbf{F}^{-1} \boldsymbol{\sigma} \mathbf{F}^{-T}$
- *Kirchoff tensor*: $\boldsymbol{\tau} \stackrel{\text{def}}{=} \det(\mathbf{F}) \boldsymbol{\sigma}$

2.3 Balance equations

This section develops the strong form of the thermo-mechanical balance principles, which are those that must be satisfied for each one of the material points of the continuum. The equations are presented as an equilibrium in both, the reference and the updated configuration. They represent the basic relations and conditions

required to the loads, stress and strain fields from the mechanical and thermodynamical continuum theory, applied to infinitesimal solid volumes ([25]).

Recalling the initial configuration of a body, \mathcal{B}_0 , in $t = 0$ and a later and deformed configuration, \mathcal{B}_t , the following balance equations are particularized to their material and spatial descriptions.

2.3.1 Conservation of mass

The mass properties of a continuum body in a configuration $\mathcal{B}_t \subset \mathbb{R}^3$, are described by means of a mass density scalar field, $\rho_t = \rho(\cdot, t) : \mathcal{B}_t \rightarrow \mathbb{R}$, so that the mass of the whole domain \mathcal{B}_t can be calculated as:

$$\text{mass}[\mathcal{B}_t] = \int_{\mathcal{B}_t} \rho(\mathbf{x}, t) dv \quad (2.12)$$

Using this definition, the *principle of conservation of mass* in the domain \mathcal{B}_t can be established. Therefore, we will not consider any chemical nor nuclear reactions that imply modification of the body mass. Also, relativistic effects will not be considered. Mathematically the principle is expressed in the following way:

$$\frac{d}{dt} \text{mass}(\mathcal{B}_t) = 0 \quad (2.13)$$

The fact that the mass does not change between the reference and deformed domains represents the basic property of the Lagrangian formulation, as opposed to the *Eulerian formulation* which establishes a *control volume* instead to formulate the equilibrium. In the case of the *control mass* of the Lagrangian formulation, the deformation of the system allows only transmission of energy through the boundary, while the total mass will be constant:

$$\text{mass}(\mathcal{B}_0) = \text{mass}(\mathcal{B}_t) \quad (2.14)$$

At the reference configuration, the total mass is known if the material density field, $\rho_0(\mathbf{X})$, is defined. The spatial density, $\rho(\mathbf{x}, t)$, will vary through the deformation but the conservation of mass implies that the infinitesimal mass is constant (and positive):

$$dm(\mathbf{X}) = dm(\mathbf{x}, t) > 0 \quad (2.15)$$

$$\rho(\mathbf{X}) dV(\mathbf{X}) = \rho(\mathbf{x}, t) dv(\mathbf{x}, t) \quad (2.16)$$

As the volume is always positive, so should be the density fields. The change in

volume is proportional to the determinant of the deformation gradient

$$dv = J dV; \quad J = \det(F(\mathbf{X}, t)) \quad (2.17)$$

so the relation between the density fields is finally:

$$\rho_m(\mathbf{X}, t) J(\mathbf{X}, t) = \rho_o(\mathbf{X}) \quad (2.18)$$

where the material description of the spatial density has been used, $\rho_m(\mathbf{X}, t) = \rho(\boldsymbol{\varphi}(\mathbf{X}, t), t)$.

2.3.2 Momentum balance

Considering a body with continuous movement and deformation in time and space \mathbb{R}^3 , let $\mathbf{v}_t = \mathbf{v}(\cdot, t) : \mathcal{B}_t \rightarrow \mathcal{V}$ the velocity field. The *linear momentum* is defined as

$$\mathbf{l}[\mathcal{B}_t] = \int_{\mathcal{B}_t} \rho(\mathbf{x}, t) \mathbf{v}(\mathbf{x}, t) dv \quad (2.19)$$

In the same way as a force can be associated to a torque, the *angular momentum* can be defined with respect to some point \mathbf{z} as the linear momentum times the perpendicular distance to that point. Mathematically, it expressed as

$$\mathbf{j}[\mathcal{B}_t]_{\mathbf{z}} = \int_{\mathcal{B}_t} (\mathbf{x} - \mathbf{z}) \times \rho(\mathbf{x}, t) \mathbf{v}(\mathbf{x}, t) dv \quad (2.20)$$

We consider that the body is subjected to a surface loading, described by a *traction field* in its boundary $\mathbf{t}_t = \mathbf{t}(\cdot, t) : \partial\mathcal{B}_t \rightarrow \mathcal{V}$, and also to a set of external volumetric loads gathered in the *volumetric field of specific loads* $\mathbf{b}_t = \mathbf{b}(\cdot, t) : \mathcal{B}_t \rightarrow \mathcal{V}$.

The *resultant force* of both fields in the whole domain \mathcal{B}_t is then

$$\mathbf{r}[\mathcal{B}_t] = \mathbf{r}_s[\mathcal{B}_t] + \mathbf{r}_b[\mathcal{B}_t] = \int_{\partial\mathcal{B}_t} \mathbf{t}(\mathbf{x}, t) da + \int_{\mathcal{B}_t} \rho(\mathbf{x}, t) \mathbf{b}(\mathbf{x}, t) dv \quad (2.21)$$

And the resultant torque around a point \mathbf{z} is

$$\boldsymbol{\vartheta}[\mathcal{B}_t] = \boldsymbol{\vartheta}_s[\mathcal{B}_t] + \boldsymbol{\vartheta}_b[\mathcal{B}_t] \quad (2.22)$$

$$\boldsymbol{\vartheta}_s[\mathcal{B}_t] = \int_{\partial\mathcal{B}_t} (\mathbf{x} - \mathbf{z}) \times \mathbf{t}(\mathbf{x}, t) da \quad (2.23)$$

$$\boldsymbol{\vartheta}_b[\mathcal{B}_t] = \int_{\mathcal{B}_t} (\mathbf{x} - \mathbf{z}) \times \rho(\mathbf{x}, t) \mathbf{b}(\mathbf{x}, t) dv \quad (2.24)$$

The *Inertia laws* can be expressed in the following way:

With respect to a reference frame, the time derivative of the linear momentum of a material body equals the resultant applied force, and the change in time of the

angular momentum about any fixed point equals the total torque around that point. Mathematically:

$$\frac{d}{dt} \mathbf{l}[\mathcal{B}_t] = \mathbf{r}[\mathcal{B}_t] \quad (2.25)$$

$$\frac{d}{dt} \mathbf{j}[\mathcal{B}_t]_z = \mathfrak{J}[\mathcal{B}_t]_z \quad (2.26)$$

A. Balance of linear momentum

The internal equilibrium of linear momentum at any point requires the inertia effects to compensate the difference of internal and external forces:

$$\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} = \rho \dot{\mathbf{v}} \quad (2.27)$$

Or, in components:

$$\sigma_{ij,j} + \rho b_i = \rho \dot{v}_i \quad (2.28)$$

When the balance is applied to material variables, a similar expression is obtained:

$$\nabla_0 \cdot \mathbf{P} + \rho_0 \mathbf{b}_m = \rho_0 \frac{\partial^2}{\partial t^2} \boldsymbol{\varphi} \quad (2.29)$$

where:

$$\mathbf{P} = \mathbf{P}(\mathbf{X}, t) = \det \mathbf{F}(\mathbf{X}, t) \boldsymbol{\sigma}(\boldsymbol{\varphi}(\mathbf{X}, t), t) \mathbf{F}(\mathbf{X}, t)^{-T},$$

is the first Piola-Kirchhoff tensor, previously defined, and

$$\mathbf{b}_m = \mathbf{b}(\boldsymbol{\varphi}(\mathbf{X}, t), t).$$

is the material description of the volumetric load field.

B. Balance of angular momentum

Considering or not the inertial effects, the angular momentum equilibrium implies the symmetry of the Cauchy stress tensor:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T \quad (2.30)$$

Or, in components:

$$\sigma_{ij} = \sigma_{ji} \quad (2.31)$$

When applied in the material description, results in the symmetry of the second Piola-Kirchhoff tensor:

$$\mathbf{P} \mathbf{F}^T = \mathbf{F} \mathbf{P}^T \quad (2.32)$$

$$\mathbf{S} = \mathbf{S}^T \quad (2.33)$$

2.3.3 Balance of mechanical energy

The integration of all the power contributions over the domain indicates the variation of the mechanical energy over time. It is a direct result from the application of the previous balance laws. The spatial description of the mechanical power balance takes the following form:

$$\frac{d}{dt}\mathcal{K}[\mathcal{B}_t] + \int_{\mathcal{B}_t} \boldsymbol{\sigma} : \mathbf{d} \, dv = \mathcal{P}_{ext}[\mathcal{B}_t] \quad (2.34)$$

where:

$$\mathcal{K}[\mathcal{B}_t] = \int_{\mathcal{B}_t} \frac{1}{2} \rho \mathbf{v} \cdot \mathbf{v} \, dV$$

is the total kinetic energy stored in the body, \mathcal{B}_t ,

$$\mathbf{d} = \nabla^{sym} \mathbf{v} = \frac{1}{2} (\text{grad } \mathbf{v} + \text{grad}^T \mathbf{v})$$

is the velocity gradient, also called *deformation rate*, and

$$\mathcal{P}_{ext}[\mathcal{B}_t] = \int_{\partial_t \mathcal{B}_t} \mathbf{v} \cdot \mathbf{t} \, dA + \int_{\mathcal{B}_t} \rho \mathbf{b} \cdot \mathbf{v} \, dV$$

is the power of external loads over \mathcal{B}_t and its boundary.

When the kinetic energy and external power are described in the reference configuration, the balance implies the definition of the internal power using material variables:

$$\frac{d}{dt}\mathcal{K}[\mathcal{B}_0] + \int_{\mathcal{B}_0} \mathbf{P} : \dot{\mathbf{F}} \, dV = \mathcal{P}_{ext}[\mathcal{B}_0] \quad (2.35)$$

where:

$$\mathcal{K}[\mathcal{B}_0] = \int_{\mathcal{B}_0} \frac{1}{2} \rho \mathbf{V} \cdot \mathbf{V} \, dV$$

$$\dot{\mathbf{F}} = \dot{\mathbf{F}}(\mathbf{X}, t) = \nabla_0 \dot{\boldsymbol{\phi}}(\mathbf{X}, t).$$

is the material derivative of the deformation gradient, and

$$\mathcal{P}_{ext}[\mathcal{B}_0] = \int_{\partial_T \mathcal{B}_0} \mathbf{V} \cdot \mathbf{t} \, dA + \int_{\mathcal{B}_0} \rho \mathbf{B} \cdot \mathbf{V} \, dV$$

2.3.4 Thermodynamics

The external thermal power, \mathcal{Q} , represents the change in heat energy produced by external loads. This power can be introduced in the system through the boundary as a heat flux, \mathbf{q}_n , or as a volumetric heat source, r . The *Stokes' heat flux theorem* establishes that the flux going through the boundary corresponds to its normal projection. As the normal is usually taken as positive in the outward direction, sign is reversed in the sum of thermal power:

$$\mathcal{Q}[\mathcal{B}_0] = \int_{\mathcal{B}_t} r \, dv - \int_{\partial_t \mathcal{B}_t} \mathbf{q} \cdot \mathbf{n} \, da \quad (2.36)$$

$$= \int_{\mathcal{B}_0} R \, dV - \int_{\partial_t \mathcal{B}_0} \mathbf{Q} \cdot \mathbf{N} \, dA \quad (2.37)$$

where:

\mathbf{q} is the Cauchy (or true) heat flux and

\mathbf{Q} is the Piola-Kirchoff heat flux tensor.

The integration of the heat fluxes over the boundary can be related between configurations using Nanson's formula:

$$\int_{\partial_t B_0} \mathbf{Q} \cdot \mathbf{N} \, dA = \int_{\partial_t B_t} \mathbf{q} \cdot \mathbf{n} \, da = \int_{\partial_t B_0} \mathbf{q} \cdot \mathbf{JF}^{-T} \mathbf{N} \, dA = \int_{\partial_t B_0} \mathbf{JF}^{-1} \mathbf{q} \cdot \mathbf{N} \, dA \quad (2.38)$$

Comparing the previous manipulation, the *Piola transformation* of the heat flux can be defined as:

$$\mathbf{Q} = \mathbf{JF}^{-1} \mathbf{q} \quad (2.39)$$

The thermal power introduced in the system produces a change in the internal energy, \mathcal{W} . Adding its contribution to the mechanical power, the first law of thermodynamics states the equilibrium of the internal energy of a continuum under thermo-mechanical loading:

$$\frac{d}{dt} \mathcal{K}(t) + \frac{d}{dt} \mathcal{W}(t) = \mathcal{P}_{ext}(t) + \mathcal{Q}(t) \quad (2.40)$$

A. Spatial description

For a thermoelastic body, the internal energy is composed of elastic and thermal energy due to volumetric and temperature variations:

$$\frac{d}{dt} \mathcal{W} = \int_{B_t} \boldsymbol{\sigma} : \mathbf{d} \, dv + \int_{B_t} \frac{d}{dt} (C\Theta) \, dv \quad (2.41)$$

where:

$C = \rho c_p$ is the volumetric heat capacity of the material, and

Θ is the temperature.

The thermo-mechanical coupling is produced mainly by thermal expansion and balanced by the external power \mathcal{P}_{ext} . Mechanical processes that imply changes in temperature are limited to thermal dissipation, which only occurs during plastic deformation [Stainer2008]. Other thermoelastic effects are almost insignificant and can be neglected.

Once this simplification is made and using the internal thermal energy, \mathcal{H} , as a state variable, the thermal equilibrium of the thermal power can be stated:

$$\frac{d}{dt} \mathcal{H}(t) = \mathcal{Q}(t) \quad (2.42)$$

$$\int_{B_t} \frac{d}{dt} (C\Theta) \, dv = \int_{B_t} r \, dv - \int_{\partial_t B_t} \mathbf{q} \cdot \mathbf{n} \, da \quad (2.43)$$

Applying the divergence theorem to the last term and localizing the equation, the strong form of the thermal power balance in the deformed configuration is obtained:

$$\frac{d}{dt}(C\Theta) = r - \nabla \cdot \mathbf{q} \quad (2.44)$$

The Second Principle of Thermodynamics states the non-negative production of entropy. Previous considerations allow to not take into account the entropy in the formulation and just reduce it to its thermal implications.

The Clausius-Planck inequality indicates that the thermal energy always flows opposing the thermal gradient:

$$-\mathbf{q} \cdot \nabla \Theta \geq 0 \quad (2.45)$$

Duhamel's law of heat conduction introduces the positive semi-definite *spatial thermal conductivity tensor*, κ , to establish the relation between the thermal gradient and flux:

$$\mathbf{q} = -\kappa \nabla \Theta \quad (2.46)$$

For isotropic materials, the thermal conductivity tensor has only one parameter and can be simplified to $\kappa = \kappa \mathbf{I}$, where \mathbf{I} is the second order unit tensor. In this case, the previous relation is known as the *Fourier's law of heat conduction*:

$$\mathbf{q} = -\kappa \nabla \Theta \quad (2.47)$$

Going back to the strong form of the balance eq. 2.44, the heat flux can now be described in terms of the temperature:

$$\frac{d}{dt}(C\Theta) = r + \nabla \cdot (\kappa \nabla \Theta) \quad (2.48)$$

B. Material description

A similar procedure can be done in the material configuration, where the balance of power takes the following form:

$$\rho C_p \frac{d\Theta}{dt} = R - \nabla_0 \cdot \mathbf{Q} \quad (2.49)$$

Taking into account the between gradients in the deformed and reference configurations, $\text{grad} \Theta = \mathbf{F}^{-T} \text{Grad} \Theta$, and eq. 2.39, Duhamel's law can be related to the reference geometry:

$$\mathbf{J}^{-1} \mathbf{F} \mathbf{Q} = -\kappa \mathbf{F}^{-T} \nabla_0 \Theta \quad (2.50)$$

Defining the material thermal conductivity tensor, $\kappa_0 = \mathbf{J} \kappa$ the expression is simplified:

$$\mathbf{Q} = -\mathbf{F}^{-1} \kappa_0 \mathbf{F}^{-T} \nabla_0 \Theta \quad (2.51)$$

If the material is thermally isotropic, using the inverse of the Cauchy-Green tensor, $\mathbf{C}^{-1} = \mathbf{F}^{-1}\mathbf{F}^{-T}$, the Fourier's law is then:

$$\mathbf{Q} = -\kappa_0 \mathbf{C}^{-1} \nabla_0 \Theta \quad (2.52)$$

Finally, the thermal power equilibrium in material coordinates is stated:

$$\rho C_p \frac{d\Theta}{dt} = R + \nabla_0 \cdot (\mathbf{F}^{-1} \kappa_0 \mathbf{F}^{-T} \nabla_0 \Theta) \quad (2.53)$$

2.4 Constitutive equations: hyperelasticity

The thermo-mechanical behaviour of the solids produces the strain state at each point to be an addition of the thermal and mechanical parts. As thermal expansion, β , is considered isotropic:

$$\mathbf{E} = \mathbf{E}^e + \beta \mathbf{I}(\Theta - \Theta_0) \quad (2.54)$$

In thermal-elastic materials, the stress field is function of the mechanical strain state and the temperature.

$$\mathbf{S} = \mathbf{S}(\mathbf{E}^e, \Theta) \quad (2.55)$$

Thermal-hyperelastic materials are characterized by the existence of a *strain energy density* function $W(\mathbf{F}, \Theta, \mathbf{X})$, such that the Piola-Kirchoff first tensor \mathbf{P} is obtained as:

$$\mathbf{P} = \frac{\partial W(\mathbf{F}^e, \Theta, \mathbf{X})}{\partial \mathbf{F}^e(\mathbf{X})} \quad (2.56)$$

The thermal dependence relies on the nonlinear material properties, which are temperature dependent. This implies that the work of the stress along a deformation is path-independent, depending only on the initial and final deformation states.

$$\int_0^1 dW = \int_{t_0}^{t_1} \mathbf{P} : \dot{\mathbf{F}}^e dt = W_{01}(\mathbf{F}^e, \Delta\Theta_0^1)$$

One of the simplest expressions for strain energy density function W is the Saint-Venant Kirchhoff law, given by:

$$W(\mathbf{E}^e, \Theta) = \frac{\lambda}{2} [\text{tr}(\mathbf{E}^e)]^2 + \mu \mathbf{E}^e : \mathbf{E}^e$$

$$\mathbf{S}(\mathbf{E}^e, \Theta) = \lambda \text{tr}(\mathbf{E}^e) \mathbf{1} + 2\mu \mathbf{E}^e$$

$$\mathbb{C} = \lambda(\mathbf{1} \otimes \mathbf{1}) + 2\mu \mathbb{I}$$

$$\mathbb{C}_{IJKL} = \lambda \delta_{IJ} \delta_{KL} + 2\mu \delta_{IK} \delta_{JL}$$

where, being $E = E(\Theta)$ the Young's Modulus and $\nu = \nu(\Theta)$ the Poisson coefficient:

$\lambda(\Theta) = \frac{E\nu}{(1+\nu)(1-2\nu)}$ is the Lamé's first parameter, and
 $\mu(\Theta) = \frac{E}{2(1+\nu)}$ is the shear modulus.

This material shows some instabilities under extremely large deformations, as its law leads to a null stress state with an infinite compression, $\lim_{J \rightarrow 0} \sigma = 0$. In those cases other hyperelasticity laws such as Neo-Hookean are used with some extra cost in terms of operations and computer memory.

2.5 Variational principles: weak formulation

The balance of linear momentum, in either spatial or material description, must be enforced at any point of the body. On the other hand, the principle of the mechanical virtual work states that the work performed by the applied forces under any *compatible virtual displacements* $\delta \mathbf{x}$ must vanish:

$$\delta \Pi_{mech} = 0$$

Applying compatible virtual temperatures to the thermal equilibrium, the principle of thermal virtual power is stated:

$$\delta \dot{\Pi}_{th} = 0$$

A more formal statement of this principle is obtained defining a set of test functions $\boldsymbol{\eta}$ such that $\boldsymbol{\eta}(\mathbf{x}) = \mathbf{0}$ in the boundary where essential conditions are applied. The weak form is obtained by weighting the strong form with these test functions, integrating by parts and applying the divergence theorem. This procedure is equivalent to the one that we present here provided that tests functions are defined in the same functional space as the spatial coordinates and temperatures, representing virtual displacements or virtual temperature variations, respectively.

2.5.1 Mechanical virtual work

The mechanical virtual work has several contributions:

$$\delta \Pi_{mech} = 0 = \delta \Pi_{inertia} + \delta \Pi_{internal} - \delta \Pi_{external} \quad (2.57)$$

where each of the terms is expressed as:

$$\delta \Pi_{inertia} = \int_{\mathcal{B}} \rho \dot{\mathbf{v}} \cdot \delta \mathbf{x} \, dV \quad (2.58)$$

$$\delta \Pi_{internal} = \int_{\mathcal{B}_0} \mathbf{P} : \delta \mathbf{F} \, dV_0 \quad (2.59)$$

$$\delta \Pi_{external} = \int_{\mathcal{B}} \mathbf{b} \cdot \delta \mathbf{x} \, dV + \int_{\partial \mathcal{B}} \mathbf{t} \cdot \delta \mathbf{x} \, dS \quad (2.60)$$

The *total Lagrangian* description of the internal virtual work is given by:

$$\delta\Pi_{internal} = \int_{B_0} (\mathbf{F} \mathbf{S}) : \delta\mathbf{F} \, dV_0 = \int_{B_0} \mathbf{S} : \delta\mathbf{E} \, dV_0 \quad (2.61)$$

Although it is also possible to express the virtual work in the deformed configuration, obtaining the following *actualized lagrangian* description:

$$\delta\Pi_{internal} = \int_B \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon} \, dV = \int_B \boldsymbol{\tau} : \delta\mathbf{e} \, dV \quad (2.62)$$

Finally, using the jacobian of the deformation and the Piola transformation from (Eq. 2.39), the external power can be also expressed in the original configuration:

$$\delta\Pi_{external} = \int_{B_0} J\mathbf{b} \cdot \delta\mathbf{x} \, dV + \int_{\partial B_0} J\mathbf{F}^{-1}\mathbf{t} \cdot \delta\mathbf{x} \, dS \quad (2.63)$$

2.5.2 Thermal virtual power

The thermal virtual power has the following contributions:

$$\delta\dot{\Pi}_{thermal} = \delta\dot{\Pi}_{capacitance} - \delta\dot{\Pi}_{external} - \delta\dot{\Pi}_{conduction} = 0 \quad (2.64)$$

where each term takes the following form:

$$\delta\dot{\Pi}_{capacitance} = \int_{B_0} \rho C_p \frac{d\Theta}{dt} \delta\Theta \, dV \quad (2.65)$$

$$\delta\dot{\Pi}_{external} = \int_{B_0} R\delta\Theta \, dV + \int_{\partial B_0} \mathbf{Q}\delta\Theta \cdot \mathbf{n} \, dS \quad (2.66)$$

$$\delta\dot{\Pi}_{conduction} = \int_{B_0} (\mathbf{F}^{-1}\boldsymbol{\kappa}_0\mathbf{F}^{-T}\nabla_0\Theta) \cdot \nabla_0\delta\Theta \, dV \quad (2.67)$$

2.5.3 Modifications of the weak form

When the test functions do not vanish at the essential boundary, $\boldsymbol{\eta}(\mathbf{x}) \neq \mathbf{0}$, an additional term must be added to the virtual work to meet this requirement:

$$\delta\bar{\Pi} = 0 = \delta\Pi + \delta\Pi_{constrains} \quad (2.68)$$

The essential boundary, ∂B_D , is subjected to a set of conditions, $\boldsymbol{\Phi}(\mathbf{x}) = \mathbf{0}$, that can be implemented by different means.

Lagrange multipliers. The contribution to the virtual work takes the following form:

$$\delta\bar{\Pi} = \delta\Pi + \int_{\partial B_D} \delta\boldsymbol{\lambda} \cdot \boldsymbol{\Phi} \, dS + \int_{\partial B_D} \boldsymbol{\lambda} \cdot \delta\boldsymbol{\Phi} \, dS \quad (2.69)$$

Penalty method does not impose exactly the constraint. Instead, it adds a force that is proportional to the constraint infringement, multiplied by a penalty factor, α :

$$\delta\bar{\Pi} = \delta\Pi + \alpha \int_{\delta\mathcal{B}_D} \mathbf{\Phi} \cdot \delta\mathbf{\Phi} dS \quad (2.70)$$

For simplicity's sake, the penalization parameter has been chosen to be constant but it can be different for each condition Φ_i . In that case it is represented by a penalization diagonal matrix $\alpha = [\alpha_{ii}]$.

Augmented Lagrange represents the combination of the previous two methods. Its based on adding the constraint violation to the Lagrange multiplier: $\lambda = \frac{1}{2}\alpha \cdot \mathbf{\Phi} + \lambda^*$:

$$\delta\bar{\Pi} = \delta\Pi + \int_{\delta\mathcal{B}_D} \delta\lambda^* \cdot \mathbf{\Phi} dS + \int_{\delta\mathcal{B}_D} (\alpha\mathbf{\Phi} + \lambda^*) \cdot \delta\mathbf{\Phi} dS \quad (2.71)$$

Meshfree Discretization of Flexible Bodies

3

3.1 Meshfree Functions

Meshfree techniques use separation of variables in a similar sense as the FEM. Inside a flexible domain, the variables, $\mathbf{u} = \{\mathbf{x}(\mathbf{X}, t), \boldsymbol{\Theta}(\mathbf{X}, t)\}$, of a point of interest—mainly a Gauss point in a Galerkin formulation or a node for collocation methods—at any time are discretized using the values of the shape functions $N(\mathbf{X})$ of each node, I , located inside the support domain of \mathbf{X} , of size SP :

$$u_i(\mathbf{X}, t) \simeq u_i^h(\mathbf{X}, t) = \sum_{I=1}^{SP} N_I(\mathbf{X}) q_{iI}(t) \quad (3.1)$$

with:

$i = \{1, \dots, m\text{-DoFs}\}$ degrees of freedom, and

$q_{iI}(t)$ generalized coordinates of the flexible body.

The main difference between finite element and meshfree discretizations is that, instead of defining the shape functions inside non-intersecting elements, the support domains for different points of interest must be overlapping.

The advantage of this approach is that the discretization does not depend on the quality of a mesh. On the other side, it can easily be seen that the computational cost will be greater because of the following aspects:

- As the domains of influence intersect with each other, the number of sums needed for describing the motion of all the points of interest of the domain will be greater than those of a FEM discretization.
- As there is no connectivity defined, the definition of the support domain needs the application of node search techniques.

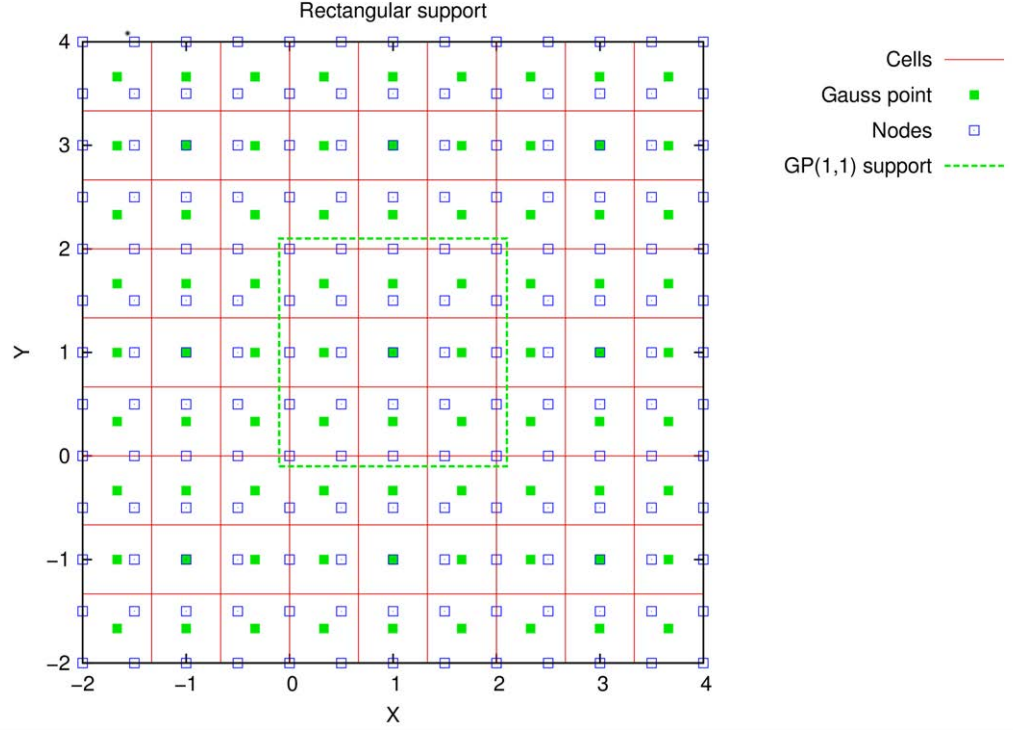


Figure 3.1: Typical 2D distribution of nodes and Gauss points with overlapping supports

- Special definition of the distance function must be done for the points of interest located near a non-convex boundary.
- The shape functions $N_j(\mathbf{X})$ usually are not analytically defined so calculating its value for each point of interest can imply the solution of a linear problem of a size equal to the support nodes number.

If a total Lagrangian formulation is used, the approximation is computed at the reference configuration. With this formalism, the evolution in time of the position of any point is given by a set of basis functions $b_i(\mathbf{X})$ and unknowns $a_i(\mathbf{X}, t)$ which are, in general, function of the material coordinates \mathbf{X} . Let a scalar function, $u(\mathbf{X}) \in C^0(\mathbb{R}^n)$ (e.g., a degree of freedom) be approximated by a discrete function:

$$u^h(\mathbf{X}) = \sum_{j=1}^m b_j(\mathbf{X}) a_j(\mathbf{X}, t) = \mathbf{b}^T(\mathbf{X}) \mathbf{a}(\mathbf{X}, t) \quad (3.2)$$

This expression is very similar to equation (3.1), but note that it just represents a shift of basis for just one of the degrees of freedom. The basis of the approximation is \mathbf{b} while \mathbf{a} represent the coefficients of the approximation in this new basis. Note that both terms now depend on the coordinate.

The type of functions that the basis $b(\mathbf{X})$ contains along with the way that the coefficients $a_j(\mathbf{X})$ are computed leads to different types of approximations.

Two commonly used approximations are presented:

- The moving least-square approximation (MLS) is used in the EFG method [9]. It uses a set of monomials as basis functions, and the coefficients are obtained by a weighed least squared fit.
- Radial Basis Functions are used in the so called RPIM method [48]. The basis is interpolated through the support points to obtain the coefficients of the shape functions.

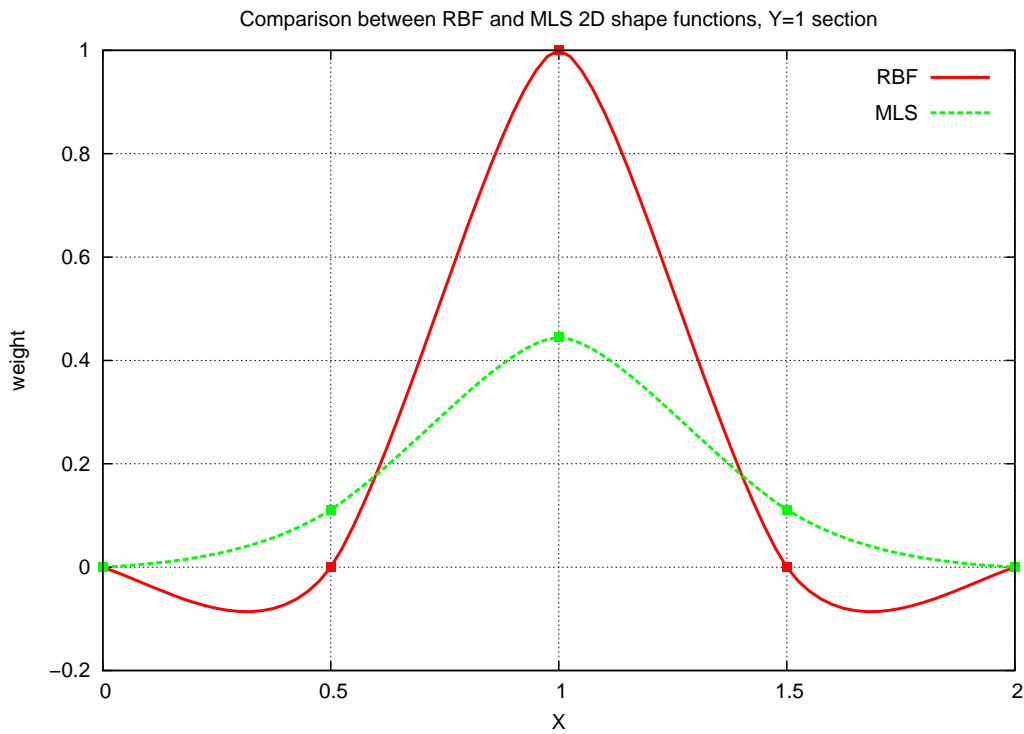


Figure 3.2: Shape functions for RBF and MLS approximations

The main difference is that only interpolation functions conserve the Kronecker delta property which states that the value of the shape function associated to a given support node is 1 at the node, and 0 at the rest of them, and it is a basic property of FEM. When it does not hold, $\mathbf{u}^h(\mathbf{X}_j) \neq \mathbf{q}_j$, and the formulation of constraints is more involved; direct elimination is not possible for simple constraints, requiring the addition of constraint equations to couple the desired degrees of freedom, even if the nodes are coincident. Note also that the delta property must be satisfied when dealing with continuous varying boundary conditions for achieving at least linear accuracy (*patch test*).

3.1.1 Properties of the meshfree shape functions

The common characteristic of all discretization methods is that they all describe a *partition of unity*:

$$\sum_{j=1}^{SP} N_j(\mathbf{X}) = 1$$

Used as weight and shape functions, meshless discrete functions ideally should have the following properties [47]:

Arbitrary nodal distribution: a reasonable approximation must be built from any regular or irregularly distributed set of points.

Stability: small differences between different set of points should produce approximations that do not blow up the numerical method.

Completeness: The shape functions must represent exactly all polynomial terms of order up to the highest derivative of the residual equation, m . A set of shape functions that satisfies this condition is called *m-complete*.

Compatibility: The shape functions should provide displacement continuity over the domain. Physically the compatibility condition insures that no material gaps appear as the numerical model deform.

Consistency: The approximation is said to have m -order consistency if it is compatible and m -complete.

Compact support: The points used for the shape function approximation must be lye inside a compact subdomain much smaller than the whole continuum to simulate.

Efficiency: The rate of precision to computing time must be in the same order as equivalent FEM discretization or better.

Delta Function Property: As the formulation of constraints is greatly simplified if the shape functions behave as the Kronecker delta function.

3.1.2 Moving Least Squares aproximation

The basis functions correspond to a set of monomials. In the case of a linear basis, we have:

$$\mathbf{b}(\mathbf{X}) = \mathbf{p}(\mathbf{X}) = \{1 \quad X_1 \quad \dots \quad X_m\}^T$$

Note that the number of coefficients to determine depends of the type of basis. If a higher order basis is chosen, the linear system that defines the approximation

values is greater. The values a_i are determined by a weighted least square fit. The weighting function is a compactly supported function (or window function). This means that given a continuous distance function $s(\mathbf{X} - \mathbf{X}_i)$, the weight function $w(s)$ must be positive inside the support domain and zero outside.

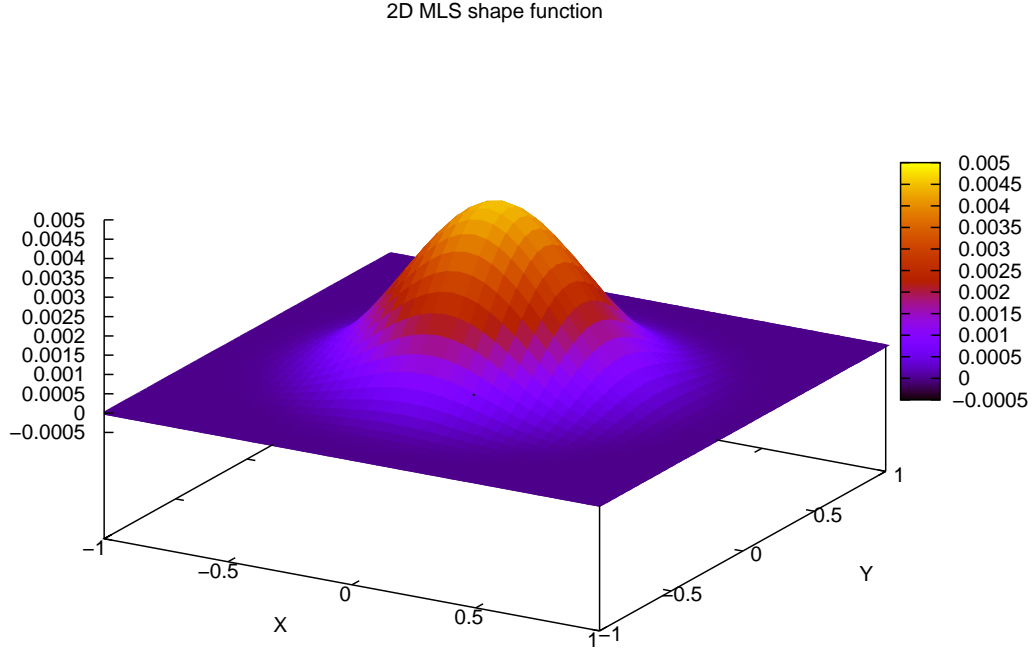


Figure 3.3: Shape functions for MLS approximations

When dealing with convex domains, an euclidean norm can be used:

$$s_i = \|\mathbf{X} - \mathbf{X}_i\|$$

but for non-convex boundaries, other functions have been proposed [7]. In any case, the weight functions are defined using the dimensionless parameter

$$\bar{s}_i = s_i / s_{max}$$

Commonly used weight functions are:

- Gaussian:

$$w(s_i) = \begin{cases} e^{-(\bar{s}_i/\alpha)^2} & \text{for } \bar{s} \leq 1 \\ 0 & \text{for } \bar{s} > 1 \end{cases} \quad (3.3)$$

where α is a constant that determines the truncation of the Gaussian, with a typical value of 0.4.

- Cubic spline:

$$w(s_i) = \begin{cases} \frac{2}{3} - 4\bar{s}_i^2 + 4\bar{s}_i^3 & \text{for } \bar{s}_i \leq \frac{1}{2} \\ \frac{4}{3} - 4\bar{s}_i + 4\bar{s}_i^2 - \frac{4}{3}\bar{s}_i^3 & \text{for } \frac{1}{2} < \bar{s}_i \leq 1 \\ 0 & \text{for } \bar{s}_i > 1 \end{cases}$$

- Quartic spline

$$w(s_i) = \begin{cases} 1 - 6\bar{s}_i^2 + 8\bar{s}_i^3 - 3\bar{s}_i^4 & \text{for } \bar{s}_i \leq 1 \\ 0 & \text{for } \bar{s}_i > 1 \end{cases}$$

Using a standard weighted least-square fit method, the next functional can be defined to estimate the error of the approximation:

$$\mathcal{J} = \sum_{j=1}^n w(s_j) \left[\sum_{i=1}^m p_i(\mathbf{X}_j) a_i(\mathbf{X}) - x_j^h \right]^2 = (\mathbf{P}\mathbf{a} - \mathbf{x}^h)^T \mathbf{W} (\mathbf{P}\mathbf{a} - \mathbf{x}^h) \quad (3.4)$$

where:

n is the number of support points.

$$\mathbf{x}^h = \{x_i^h\}$$

$$\mathbf{P} = [p_{ij}] = [p_j(\mathbf{X}_i)] = [\mathbf{p}(\mathbf{X}_i)^T]$$

$$\mathbf{W}(\mathbf{X}) = [w_{ii}] = [w(s_j(\mathbf{X}))]$$

Minimizing the functional (3.4), we get the residual form:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{a}} = \mathbf{A}(\mathbf{X})\mathbf{a}(\mathbf{X}) - \mathbf{B}(\mathbf{X})\mathbf{q} = 0 \quad (3.5)$$

where:

$$\mathbf{A} = \mathbf{P}^T \mathbf{W}(\mathbf{X}) \mathbf{P} = \sum_{k=1}^n w(\mathbf{X} - \mathbf{X}_k) \mathbf{p}(\mathbf{X}_k) \mathbf{p}(\mathbf{X}_k)^T \longrightarrow A_{ij} = \sum_{k=1}^n w_k P_{ki} P_{kj}$$

$$\mathbf{B} = \mathbf{P}^T \mathbf{W}(\mathbf{X}) = [w(\mathbf{X} - \mathbf{X}_j) \mathbf{p}(\mathbf{X}_j)] \longrightarrow B_{ij} = w_j P_{ji}$$

Equation (3.5) provides the values for the coefficients a_i :

$$\mathbf{a}(\mathbf{X}) = \mathbf{A}^{-1}(\mathbf{X}) \mathbf{B}(\mathbf{X}) \mathbf{x}^h$$

What is more, as the approximate coordinate value is given by:

$$x_i^h(\mathbf{X}) = \mathbf{p}^T(\mathbf{X}) \mathbf{a}(\mathbf{X}) \quad ,$$

the shape functions are defined using equation (3.1) as:

$$\mathbf{N} = [N_I] = \mathbf{p}^T(\mathbf{X}) \mathbf{A}^{-1}(\mathbf{X}) \mathbf{B}(\mathbf{X}) \quad (3.6)$$

Note that \mathbf{N} is defined as a row matrix, so using each column of the matrix $\mathbf{B} = [\mathbf{B}_I]$ such that $(b_I)_i = w_I p_i$, each shape function value can be computed as:

$$N_I = \mathbf{p}^T \mathbf{A}^{-1} \mathbf{B}_I \quad (3.7)$$

As stated in [7], the derivatives can be computed faster if an auxiliary variable α is defined:

$$\alpha^T = \mathbf{p}^T \mathbf{A}^{-1} \rightarrow \mathbf{A} \alpha = \mathbf{p} \quad (3.8)$$

Then, replacing (3.8) in (3.6), those partial derivatives involved in the continuum formulation are computed as:

$$\mathcal{N} = [\mathbf{N}_{,i}] = \alpha_{,i}^T \mathbf{B} + \alpha^T \mathbf{B}_{,i}$$

where \mathcal{N} is a rectangular matrix where each row represents the i -th derivative of each shape function:

$$\mathcal{N}_{iI} = N_{I,i} = \alpha_{,i}^T \mathbf{B}_I + \alpha^T \mathbf{B}_{I,i}$$

Now $\alpha_{,i}$ can be computed in an efficient manner, as in a LU decomposition, where matrix \mathbf{A} is only decomposed once:

$$\mathbf{A} \alpha_{,i} = \mathbf{p}_{,i} - \mathbf{A}_{,i} \alpha$$

Note that the decomposition of matrix \mathbf{A} is used for solving the shape functions too, as shown in equation (3.8).

3.1.3 Radial basis functions interpolation

As opposed to the MLS approximation, the RBF are interpolation functions. This means that they have the delta function property. The most attractive RBF is the Multi-quadrics (MQ) due to its high convergence rate, but other radial functions are also found in the literature. The selection of the parameters to be used involve the consideration of the size of the window function and a characteristic length parameter. The reference equations and the influence of their parameters are studied in [38].

Recalling equation (3.2), the approximation can be defined as:

$$x^h(\mathbf{X}) = \mathbf{b}^T(\mathbf{X}) \mathbf{a}(\mathbf{X}) = \{\mathbf{R}(\mathbf{X}) \quad \mathbf{p}(\mathbf{X})\}^T \left\{ \mathbf{a}_R(\mathbf{X}) \quad \mathbf{a}_p(\mathbf{X}) \right\} \quad , \quad (3.9)$$

where the term $\mathbf{R}^T(\mathbf{X}) \mathbf{a}_R(\mathbf{X})$ represents the approximation on the RBF basis, with a size which is equal to the number of support points. $\mathbf{p}^T(\mathbf{X}) \mathbf{a}_p(\mathbf{X})$ is an optional polynomial basis that can be used for the required reproducing conditions.

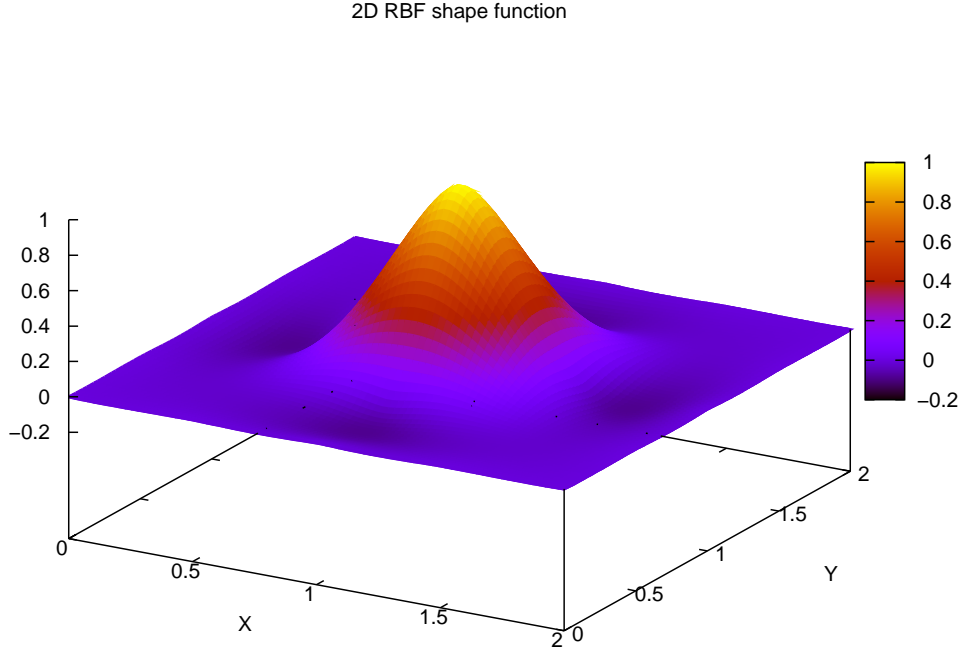


Figure 3.4: Shape functions for RBF interpolants

Multiquadrics. The original form of this RBF [19] depends only on one parameter, $c > 0$, with length units:

$$R_i = R(s_i) = \sqrt{s_i^2 + c^2}$$

The *generalized MQ* basis function adds an exponential parameter [40], $\beta = -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \dots$, although other parameters have been successfully tested, as:

$$R_i = R(s_i) = (s_i^2 + c^2)^\beta$$

G. R. Liu and co-workers [47] use a dimensionless approach for the function constant, arriving to different *optimal* values in Galerkin methods by the study of benchmark static problems:

$$R_i = R(s_i) = (s_i^2 + (\alpha_c d_c)^2)^q$$

Comparing the above equations, it can be seen that the parameter they employ is simply defined as $\alpha_c = \frac{c}{d_c}$. Also, they rename $\beta = q$ and recommend the value $q = 0.98$ or $q = 1.03$, because the moment matrix used in the interpolation fit is singular for intermediate values.

Gaussian. This RBF has an expression with a high statistical meaning, with a format which is expressed in terms of its *standard deviation*, σ :

$$R_i = R(s_i) = e^{-\frac{s_i^2}{2\sigma^2}}$$

Once again, G.R. Liu [47] propose this dimensionless form:

$$R_i = R(s_i) = e^{-\alpha_c \left(\frac{s_i}{d_c}\right)^2}$$

where they have defined a parameter α_c that weights the ratio of the distance over the average nodal separation. This parameter relates to the standard deviation by:

$$\alpha_c = \frac{d_c^2}{2\sigma^2} \quad (3.10)$$

Comparing expression (3.3) with (3.10), taking into account that they define the parameter $\alpha = \sqrt{2}\sigma$, it turns out that:

$$\alpha_c = \frac{d_c^2}{\alpha^2}$$

Polyharmonic Splines. They represent a family of exponential surfaces. The use of a low order polynomials for augmenting the interpolation basis improves the interpolation near the boundary. The general form of these functions is:

$$R_i = R(s_i) = \begin{cases} s_i^k & \text{with } k = 1, 3, 5, \dots \\ s_i^k \log(s_i) & \text{with } k = 2, 4, 6, \dots \end{cases}$$

The formulation of the splines with the logarithmic term may be changed in the implementation in order to avoid its singularity problem around $s_i = 0$:

$$R_i = R(s_i) = \begin{cases} s_i^{k-1} \log(s_i^{s_i}) & \text{for } s_i < 1 \\ s_i^k \log(s_i) & \text{for } s_i \geq 1 \end{cases}$$

When $k = 2$, the function represents a *thin plate spline* (TPS), which is the similar to the cubic spline for higher dimensions. It represents in three dimensions the bending of a thin sheet surface that goes through the interpolation points.

$$R_i = R(s_i) = s_i^2 \log(s_i)$$

In order to use RBFs with a set of scattered points, let us define first the distance

from a point of the set j to the reference point i :

$$r_k = \|\mathbf{X}_k - \mathbf{X}_i\|$$

The computation of the coefficients of the basis is done by an interpolation procedure, so the functions are forced to pass through the nodal values:

$$\mathbf{q} = \mathbf{R}_0 \mathbf{a}_R + \mathbf{P}_m \mathbf{a}_p \quad (3.11)$$

where:

$$\mathbf{R}_0 = [R_j(r_i)]_{(n \times n)}$$

$$\mathbf{P}_m = [p_j(\mathbf{X}_i)]_{(n \times m)}$$

In order to obtain the equations that must be solved to obtain the m variables incorporating the augmented basis, orthogonal conditions are added. The final system, using generalized coordinates, yields:

$$\tilde{\mathbf{q}} = \begin{Bmatrix} \mathbf{q} \\ \mathbf{0} \end{Bmatrix} = \begin{bmatrix} \mathbf{R}_0 & \mathbf{P}_m \\ \mathbf{P}_m^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a}_R \\ \mathbf{a}_p \end{Bmatrix} = \mathbf{G} \mathbf{a}$$

Solving for coefficients \mathbf{a} and substituting in equation (3.9) we obtain:

$$x^h(\mathbf{X}) = \{\mathbf{R}^T(\mathbf{X}) \quad \mathbf{p}^T(\mathbf{X})\} \mathbf{G}^{-1} \tilde{\mathbf{q}}$$

So, the augmented shape functions are expressed as:

$$\tilde{\mathbf{N}}(\mathbf{X}) = \{\mathbf{R}^T(\mathbf{X}) \quad \mathbf{p}^T(\mathbf{X})\} \mathbf{G}^{-1}$$

However, they are more efficiently computed solving the linear system:

$$\mathbf{G}^T \tilde{\mathbf{N}}(\mathbf{X}) = \{\mathbf{R}(\mathbf{X}) \quad \mathbf{p}(\mathbf{X})\}^T \quad (3.12)$$

What is more, spatial derivatives are easily calculated by:

$$x_{,l}^h(\mathbf{X}) = \tilde{\mathbf{N}}_{,l}(\mathbf{X}) \tilde{\mathbf{q}} \quad (3.13)$$

where the derivative of the shape functions is computed solving the following linear system:

$$\mathbf{G}^T \tilde{\mathbf{N}}_{,l}(\mathbf{X}) = \{\mathbf{R}_{,l}(\mathbf{X}) \quad \mathbf{p}_{,l}(\mathbf{X})\}^T \quad (3.14)$$

3.2 Bubnov-Galerkin discretization of flexible bodies

In an thermal-hyperelastic body, the strong form of the governing dynamic equations can be stated as general initial value problem. Recalling the mechanical (2.35) and thermal (2.53) equilibrium equations:

$$\left\{ \begin{array}{l} M(\ddot{\mathbf{x}}) + D(\mathbf{x}, \Theta) - f(\mathbf{x}, \dot{\mathbf{x}}, t) = 0 \\ C(\dot{\Theta}) + H(\mathbf{x}, \Theta) - Q(\mathbf{x}, t) = 0 \end{array} \right\} \text{ in the domain } \mathcal{B} \quad (3.15)$$

$$\left\{ \begin{array}{l} B(\mathbf{x}) - g(t) = 0 \\ L(\Theta) - q(\mathbf{x}, t) = 0 \end{array} \right\} \text{ in the Neumann boundary } \partial\mathcal{B}_N$$

$$\left\{ \begin{array}{l} u(\mathbf{x}, \dot{\mathbf{x}}, t) = 0 \\ T(\mathbf{x}, \Theta, t) = 0 \end{array} \right\} \text{ in the Dirichlet boundary } \partial\mathcal{B}_D$$

$$\left\{ L(\Theta) - q(\mathbf{x}, \Theta, t) = 0 \right\} \text{ in the Robin boundary } \partial\mathcal{B}_R$$

with $\{\mathbf{x}, \dot{\mathbf{x}}, \Theta\}^T = \{\mathbf{x}_0, \dot{\mathbf{x}}_0, \Theta_0\}^T$ in t_0 , and where:

D is a first order elliptic operator,

B, H, L are first order differential operators,

f, Q are the body loads and internal heat generation, respectively,

M, C are algebraic operators,

g, q is the traction field and heat flux values on the non-constrained boundary, and

u, T are the displacement and temperature constraints.

The shape functions could be partitioned for separating the mechanical (D) and thermal (T) components:

$$\mathbf{u}_i^h(\mathbf{X}, t) = \begin{Bmatrix} \mathbf{x}_i^h \\ \Theta_i^h \end{Bmatrix} = \sum_{I=1}^{SP} N_I(\mathbf{X}) \mathbf{q}_{iI}(t) \quad (3.16)$$

with

$$N_I = \begin{bmatrix} N_{DI} & \mathbf{0} \\ \mathbf{0} & N_{TI} \end{bmatrix} \quad (3.17)$$

and

$$\mathbf{q}_{iI}(t) = \begin{Bmatrix} q_{DI} \\ q_{TI} \end{Bmatrix}, \quad (3.18)$$

although they are usually the same for each degree of freedom:

$$\mathbf{u}_i^h(\mathbf{X}, t) = \begin{Bmatrix} \mathbf{x}_i^h \\ \Theta_i^h \end{Bmatrix} = \sum_{I=1}^{SP} N_I(\mathbf{X}) \mathbf{q}_{iI}(t) \quad (3.19)$$

The spatial coordinates \mathbf{x} and temperatures Θ of the body are now approximated by a set of discrete values:

$$\begin{Bmatrix} \mathbf{x}^h \\ \Theta^h \end{Bmatrix} = \begin{bmatrix} \mathbf{N}_D & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_T \end{bmatrix} \begin{Bmatrix} \mathbf{q}_D \\ \mathbf{q}_T \end{Bmatrix} = \mathbf{N} \mathbf{q}_f \quad (3.20)$$

in terms of the flexible body generalized coordinates \mathbf{q}_f and shape functions \mathbf{N} . In this section we will assume that the Dirichlet boundary conditions are exactly satisfied by the approximation. We will see later on how to enforce those conditions when the latter statement does not hold. Equation (3.15) will not hold exactly at every point of the domain when the approximation is introduced, and the difference between the exact and the approximate solutions can be stated in a residual form as:

$$\mathcal{R}_D(\mathbf{x}^h, \Theta^h) = M(\ddot{\mathbf{x}}^h) + D(\mathbf{x}^h, \Theta^h) - f(\mathbf{x}^h, \dot{\mathbf{x}}^h, t) \quad (3.21)$$

$$\mathcal{R}_T(\mathbf{x}^h, \Theta^h) = C(\dot{\Theta}^h) + H(\mathbf{x}^h, \Theta^h) - Q(\mathbf{x}^h, t) \quad (3.22)$$

The $(n + m)$ equations needed to find the q_i coefficients are built up from a set of weighted integrals:

$$\int_{B_i} \mathcal{W}_i(\mathbf{x}) \mathcal{R}_D(\mathbf{x}^h, \Theta^h) dV = 0 \quad , \quad \text{for } i = 1, 2, \dots, n \quad (3.23)$$

$$\int_{B_j} \mathcal{W}_j(\Theta) \mathcal{R}_T(\mathbf{x}^h, \Theta^h) dV = 0 \quad , \quad \text{for } j = 1, 2, \dots, m \quad (3.24)$$

The weights $\mathcal{W} = \{\mathcal{W}_1, \dots, \mathcal{W}_n, \mathcal{W}_{n+1}, \dots, \mathcal{W}_{n+m}\}^T$ are the discrete counterpart of the *test functions* or *virtual displacements*. If they are chosen so that they are the same as the *trial (or shape) functions*, then the method is called *Bubnov-Galerkin*.

$$\mathcal{W} = \begin{Bmatrix} \delta \mathbf{x}^h \\ \delta \Theta^h \end{Bmatrix} = \frac{\partial \{\mathbf{x}^h, \Theta^h\}^T}{\partial \mathbf{q}_f} \delta \mathbf{q}_f = \mathbf{N} \cdot \delta \mathbf{q}_f \quad (3.25)$$

What is more, as shape functions do not depend on time, the n -th time derivative of the spatial coordinates and temperatures is:

$$\frac{d^n}{dt^n} \{\mathbf{x}^h, \Theta^h\}^T = \begin{bmatrix} \mathbf{N}_D & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_T \end{bmatrix} \frac{d^n}{dt^n} \mathbf{q}_f = \mathbf{N} \frac{d^n}{dt^n} \mathbf{q}_f \quad (3.26)$$

Introducing the discretization (3.25) in the mechanical virtual work equations, with inertial (2.58), internal (2.59) and external (2.60) terms; and using the discretized time derivation formula, (3.26), the expressions for the discretized virtual work can be obtained. Note that the discretized temperature degree of freedom is only implicitly used through the material's properties and the thermal dilatation:

$$\delta\Pi_{mech}^h = \delta\Pi_{inertia}^h + \delta\Pi_{internal}^h - \delta\Pi_{external}^h = 0 \quad (3.27)$$

$$\delta\Pi_{inertia}^h = \delta\mathbf{q}_D^T \cdot \left[\int_{B_0} \rho \mathbf{N}_D^T \cdot \mathbf{N}_D dV \right] \cdot \ddot{\mathbf{q}}_D = \delta\mathbf{q}_D^T \cdot \mathbf{M}_D \cdot \ddot{\mathbf{q}}_D \quad (3.28)$$

$$\delta\Pi_{internal}^h = \delta\mathbf{q}_D^T \cdot \mathbf{f}_{int}^h \quad (3.29)$$

$$\delta\Pi_{external}^h = \delta\mathbf{q}_D^T \cdot \mathbf{f}_{ext}^h \quad (3.30)$$

where:

\mathbf{M}_D is the mass matrix of the flexible body,

\mathbf{f}_{int}^h is the discretized internal forces vector:

$$(\mathbf{f}_{int}^h)_I = \int_{B_0} \nabla_0 \mathbf{N}_{DI} \cdot \mathbf{P} dV_0 \text{ in a total Lagrangian formulation,}$$

$$(\mathbf{f}_{int}^h)_i = \int_B \nabla \mathbf{N}_{Di} \cdot \boldsymbol{\sigma} dV \text{ in an updated lagrangian formulation, and}$$

\mathbf{f}_{ext}^h is the discretized external forces vector, where each nodal vector is defined by:

$$(\mathbf{f}_{ext}^h)_I = \int_{B_0} \mathbf{N}_{DI} \cdot J \mathbf{b} dV + \int_{\partial B_0} \mathbf{N}_{DI} \cdot J \mathbf{F}^{-1} \mathbf{t} dS \text{ in a total Lagrangian formulation,}$$

$$(\mathbf{f}_{ext}^h)_I = \int_B \mathbf{N}_{Di} \cdot \mathbf{b} dV + \int_{\partial B} \mathbf{N}_{Di} \cdot \mathbf{t} dS \text{ in an updated lagrangian formulation.}$$

Inserting the equations (3.27), (3.29) and (3.30) into (2.57) we obtain:

$$\delta\mathbf{q}_D^T (\mathbf{M}_D \ddot{\mathbf{q}}_D + \mathbf{f}_{int}^h - \mathbf{f}_{ext}^h) = 0 \quad \forall \delta\mathbf{q}_D \text{ compatible,}$$

which leads to the following system of ordinary differential equations, provided all generalized coordinates are compatible:

$$\mathbf{M}_D \ddot{\mathbf{q}}_D + \mathbf{f}_{int}^h - \mathbf{f}_{ext}^h = \mathbf{0} \quad (3.31)$$

The thermal power is discretized by similar means:

$$\delta\dot{\Pi}_{thermal}^h = \delta\dot{\Pi}_{capacitance}^h - \delta\dot{\Pi}_{external}^h - \delta\dot{\Pi}_{conduction}^h = 0 \quad (3.32)$$

$$\delta\dot{\Pi}_{capacitance}^h = \delta\mathbf{q}_T^T \cdot \left[\int_{B_0} \rho C_p \mathbf{N}_T^T \cdot \mathbf{N}_T dV \right] \cdot \dot{\mathbf{q}}_T = \delta\mathbf{q}_T^T \cdot \mathbf{C} \cdot \dot{\mathbf{q}}_T \quad (3.33)$$

$$\delta\dot{\Pi}_{external}^h = \delta\mathbf{q}_T^T \cdot \mathbf{Q}_{ext}^h \quad (3.34)$$

$$\delta\dot{\Pi}_{conduction}^h = \delta\mathbf{q}_T^T \cdot \mathbf{H} \cdot \mathbf{q}_T \quad (3.35)$$

where:

\mathbf{C} is the capacitive matrix of the flexible body,

\mathbf{Q}_{ext}^h is the discretized external heat loads vector, with each component described by:

$$\begin{aligned} (\mathbf{Q}_{ext}^h)_I &= \int_{B_0} \mathbf{N}_{DI} \cdot \mathbf{R} \, dV + \int_{\partial B_0} \mathbf{N}_{DI} \cdot \mathbf{Q} \cdot \mathbf{n} \, dS \text{ in a total Lagrangian formulation,} \\ (\mathbf{Q}_{ext}^h)_I &= \int_B \mathbf{N}_{DI} \cdot \mathbf{r} \, dV + \int_{\partial B} \mathbf{N}_{DI} \cdot \mathbf{q} \cdot \mathbf{n} \, dS \text{ in an updated lagrangian formulation,} \\ &\text{and} \end{aligned}$$

\mathbf{H} is the discretized thermal conductivity matrix of the body:

$$\begin{aligned} \mathbf{H} &= \int_{B_0} \nabla_0 \mathbf{N}_T^T \mathbf{F}^{-1} \boldsymbol{\kappa}_0 \mathbf{F}^{-T} \nabla_0 \mathbf{N}_T \, dV \text{ in a total Lagrangian formulation,} \\ \mathbf{H} &= \int_B \nabla \mathbf{N}_T^T \boldsymbol{\kappa} \nabla \mathbf{N}_T \, dV \text{ in an updated Lagrangian formulation,} \end{aligned}$$

Once again, making the substitution of the equations (3.32), (3.34) and (3.35) into (2.64) we obtain:

$$\delta \mathbf{q}_T^T (\mathbf{C} \dot{\mathbf{q}}_T - \mathbf{H} \mathbf{q}_T - \mathbf{Q}_{ext}^h) = 0 \quad \forall \delta \mathbf{q}_T \text{ compatible,}$$

leading to the thermal system of ordinary differential equations, with the thermal generalized coordinates compatible:

$$\mathbf{C} \dot{\mathbf{q}}_T - \mathbf{H} \mathbf{q}_T - \mathbf{Q}_{ext}^h = 0 \quad (3.36)$$

3.2.1 Constrained Bubnov-Galerkin form

Lagrange multipliers have the advantage of imposing exactly the constraints although they increase the number of unknowns, lead to a DAE system and the tangent matrix is no longer diagonally dominant. Also, the system of equations turn out to be singular if the number of Lagrange multipliers is too large and their interpolation functions are not chosen carefully [11]. Penalty functions do not impose exactly the constraints and increase the condition number of the tangent matrix. Augmented Lagrange method solve previous inconveniences by adding another loop to the solver, increasing significantly the computation time.

Recalling the continuous description of the modified variational problem of section 2.5.3, the discretized constrained problem is defined next for each type of method.

Lagrange multipliers.

$$\delta \bar{\Pi} = \delta \Pi + \int_{\delta B_D} \delta \boldsymbol{\lambda} \cdot \boldsymbol{\Phi} \, dS + \int_{\delta B_D} \boldsymbol{\lambda} \cdot \delta \boldsymbol{\Phi} \, dS \quad (3.37)$$

The lagrange multipliers are approximated along the boundary by a set of functions $\lambda(\mathbf{x}, t) \simeq \sum_{i=1}^l \lambda_i(t) \phi_i^L(\mathbf{X})$. The simplest way to integrate the constrained boundary is to apply point collocation, $\phi_i^L(X) = \delta(x - x_i^L)$, where $\{x_i^L\}$ are the nodes of the essential boundary domain. Finally, this procedure leads to the following discrete residual form:

$$\delta \bar{\Pi}^h = \delta \Pi^h + \delta \mathbf{q}^T (\mathbf{\Phi}_q^T \boldsymbol{\lambda}) + \delta \boldsymbol{\lambda}^T \mathbf{\Phi} = 0 \quad (3.38)$$

where the $\{\cdot\}_q$ denotes the gradient: $\mathbf{\Phi}_q = \frac{\partial \mathbf{\Phi}}{\partial \mathbf{q}}$

The constraint force vector for each flexible body takes the form $(\mathbf{f}_{\Phi})_i = \mathbf{\Phi}_q^T \boldsymbol{\lambda}$ and the constraint set $\mathbf{\Phi}$ must be appended to the differential equation system (4.112), adding one variable for each constraint and require special time integration procedures. The resulting index-3 DAE must be solved for $(\mathbf{q}(t), \boldsymbol{\lambda}(t))$.

Penalty method

$$\delta \bar{\Pi} = \delta \Pi + \alpha \int_{\delta B_D} \mathbf{\Phi} \cdot \delta \mathbf{\Phi} dS \quad (3.39)$$

$$\delta \bar{\Pi}^h = \delta \Pi^h + \delta \mathbf{q}^T (\mathbf{\Phi}_q^T \alpha \mathbf{\Phi}) = 0 \quad (3.40)$$

If the value of this parameter is too high, it will produce an ill conditioned set of equations. The constraint force takes the form $\mathbf{f}_{\Phi} = \mathbf{\Phi}_q^T \alpha \mathbf{\Phi}$ and the differential system (4.112) completely defines the motion.

Augmented Lagrange

$$\delta \bar{\Pi} = \delta \Pi + \int_{\delta B_D} \delta \boldsymbol{\lambda}^* \cdot \mathbf{\Phi} dS + \int_{\delta B_D} (\alpha \mathbf{\Phi} + \boldsymbol{\lambda}^*) \cdot \delta \mathbf{\Phi} dS \quad (3.41)$$

The integration of this equation by collocation is equivalent to make the corresponding substitution in Eq. (3.38) and taking into account that $\delta \boldsymbol{\lambda}^T \mathbf{\Phi} = 0$; resulting:

$$\delta \bar{\Pi}^h = \delta \Pi^h + \mathbf{\Phi}_q^T (\alpha \cdot \mathbf{\Phi}) + \mathbf{\Phi}_q^T \boldsymbol{\lambda}^* = 0 \quad (3.42)$$

The system is solved by an iterative procedure where the multilpliers can be updated adding the penalty load:

$$\boldsymbol{\lambda}_{i+1}^* = \boldsymbol{\lambda}_i^* + \alpha \cdot \mathbf{\Phi}_{i+1} \quad (3.43)$$

Although another iterative loop is added to the nonlinear solving procedure, the convergence to almost the exact constraint imposition is very fast.

3.2.2 Transformation method

The transformation method [15] is the best way to impose directly the essential boundary conditions [45] but requires the inversion of the transformation matrix. As a total Lagrange formulation is used for the modeling of flexible bodies, this matrix must only be computed once, reducing the impact on the total computation cost.

The recovery of the delta property for a node can be seen as the imposition of an essential boundary condition using point collocation. Supposing that each boundary condition, $\Phi_i = 0$, is an explicit expression of the nodal displacements, each condition can then be expressed as $\Phi_i = \mathcal{F}_i - x_i = 0$, which can be put in the following form:

$$x_i(\mathbf{X}_J, t) = \mathcal{F}_i(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n, t) = g_i(\mathbf{X}_J, t) \quad (3.44)$$

Materializing the shape function of Eq. (3.1) at a constrained node J and taking into account that its variation must vanish, the following expressions have to be satisfied:

$$x_i^h(\mathbf{X}_J, t) = \sum_{I=1}^{NP} \phi_I(\mathbf{X}_J) q_{iI}(t) = g_i(\mathbf{X}_J, t) \quad (3.45)$$

$$\delta x_i^h(\mathbf{X}_J, t) = 0 \quad (3.46)$$

Using implicit addition, Eq. (3.45) can be expressed in the following way:

$$x_{ij}^h(t) = \phi_{IJ} q_{iI}(t) = g_{ij}(t) \quad (3.47)$$

changing the index for convenience, the generalized coordinates will be

$$q_{iI}(t) = \phi_{IK}^{-T} x_{iK}^h(t) \quad (3.48)$$

and substituting in the shape function definition (Eq. 3.1)

$$x_i^h(\mathbf{X}, t) = \sum_{I=1}^{NP} \phi_I(\mathbf{X}) q_{iI}(t) = \sum_{I=1}^{NP} \phi_I(\mathbf{X}) \phi_{IK}^{-T} x_{iK}^h(t) \quad (3.49)$$

The modified shape function is

$$\hat{\phi}(\mathbf{X}) = \sum_{I=1}^{NP} \phi_{KI}^{-1} \phi_I(\mathbf{X}) \quad (3.50)$$

and has recovered the delta property

$$\hat{\phi}_K(\mathbf{X}_J) = \sum_{I=1}^{NP} \phi_{KI}^{-1} \phi_I(\mathbf{X}_J) = \phi_{KI}^{-1} \phi_{IJ} = \delta_{KJ} \quad (3.51)$$



Figure 3.5: Blending meshfree approximation functions with FEM shape functions [18]

while preserving the properties of the original shape function. Note that this transformation is only necessary for the shape functions of those points that have one or more constrained nodes inside their support domain.

Flexible Multibody Systems Formulation

4

Previous chapters have described the dynamics of single flexible bodies. The reason to study more than one of those bodies in the same system is the existence of some kind of interaction between them. The present chapter presents the formulation of these systems, considering also perfectly rigid bodies, which are those where the strain can be neglected, allowing a simplified formulation of their dynamics.

Lets recall the deformation gradient of a body, $F = \nabla_0 \mathbf{x}$. For each deformation tensor, there exists a right polar decomposition into an orthogonal and a positive definite symmetric tensor: $F = R \cdot U$, representing a *stretch*, U , followed by a *rotation*, R . A rigid body motion can be defined by a deformation without stretch, $F_{rigid} = R$. This is equivalent to the condition of null strain, as the nonlinear strain measures defined in Section 2.1 depend directly on the right or left stretch tensors: $U^2 = F^T F$ and $V^2 = F F^T$, respectively.

As all the points of a rigid body will remain at the same distance between them, the motion of the body can be described exactly by a discrete set of parameters. This number defines the degrees of freedom and depends on the kind of body and the euclidean space dimension considered on a simulation.

If the number of parameters chosen to describe a rigid body is higher than its degrees of freedom, a set of constraints must be imposed to compensate this excess of variables. In a similar manner, the physical joints between the bodies reduce the degrees of freedom of the whole system, which mathematically implies also the application of constraint equations to parameters of different bodies. The total number of parameters minus all the internal and joint constraints give the degrees of freedom of the whole multibody system.

4.1 Balance equations of rigid bodies

Lets consider a general rigid body composed by d continuous and p discrete masses, with the only condition that their position is fixed in a local reference frame. The application of the balance equations for flexible bodies of Section 2.3 to rigid bodies is done by applying the previous non-stretching condition:

Conservation of mass: the mass of the domain, $\frac{d}{dt}\text{mass}[\mathcal{B}_t] = 0$, remains constant.

As the volume of the distributed mass does not change at any point, $J = 1$, the density field must also be constant: $\rho_m(\mathbf{X}, t) = \rho_o(\mathbf{X})$

Linear momentum balance: the lack of strain implies that any applied load produces directly a change in the linear momentum of the whole body:

$$\sum_{i=1}^{N_b} \int_{\mathcal{B}_i} \mathbf{b}_i dV + \sum_{j=1}^{N_t} \int_{\partial \mathcal{B}_j} \mathbf{t}_j dA + \sum_{k=1}^{N_f} \mathbf{f}_k = \sum_{l=1}^{N_d} \left[\int_{\mathcal{B}_l} \rho_l \frac{\partial^2 \mathbf{x}}{\partial t^2} dV \right] + \sum_{h=1}^{N_p} m_h \frac{\partial^2 \mathbf{x}_h}{\partial t^2} \quad (4.1)$$

Angular momentum balance: the resultant torque of the applied loads around a point \mathbf{z} is:

$$\mathfrak{D}[\mathcal{B}_t]_{\mathbf{z}} = \mathfrak{D}_f + \mathfrak{D}_t + \mathfrak{D}_b \quad (4.2)$$

$$\mathfrak{D}_f[\mathcal{B}]_{\mathbf{z}} = \sum_{i=1}^{N_f} (\mathbf{x}_i - \mathbf{z}) \times \mathbf{f}_i(\mathbf{x}, t) \quad (4.3)$$

$$\mathfrak{D}_t[\mathcal{B}]_{\mathbf{z}} = \sum_{j=1}^{N_t} \int_{\partial \mathcal{B}_j} (\mathbf{x} - \mathbf{z}) \times \mathbf{t}_j(\mathbf{x}, t) da \quad (4.4)$$

$$\mathfrak{D}_b[\mathcal{B}]_{\mathbf{z}} = \sum_{k=1}^{N_b} \int_{\mathcal{B}_k} (\mathbf{x} - \mathbf{z}) \times \mathbf{b}_k(\mathbf{x}, t) dv \quad (4.5)$$

This applied torque varies the inertial angular momentum of the body:

$$\mathfrak{D}[\mathcal{B}_t]_{\mathbf{z}} = \sum_{i=1}^{N_d} \left[\int_{\mathcal{B}_i} \rho_i \frac{\partial \mathbf{x}}{\partial t} \times (\mathbf{x} - \mathbf{z}) dV \right] + \sum_{j=1}^{N_p} m_j \frac{\partial \mathbf{x}_j}{\partial t} \times (\mathbf{x}_j - \mathbf{z}) \quad (4.6)$$

Balance of mechanical energy: The equilibrium is equivalent to the one expressed by (Eq. 2.34) neglecting the deformation energy:

$$\frac{d}{dt} \mathcal{K}[\mathcal{B}_t] = \mathcal{P}_{ext}[\mathcal{B}_t] \quad (4.7)$$

where:

$$\mathcal{K}[\mathcal{B}_t] = \sum_{i=1}^{N_d} \int_{\mathcal{B}_i} \frac{1}{2} \rho_i \mathbf{v} \cdot \mathbf{v} dV + \sum_{j=1}^{N_p} \frac{1}{2} m_j \mathbf{v}_j \cdot \mathbf{v}_j$$

$$\mathcal{P}_{ext}[\mathcal{B}_t] = \sum_{i=1}^{N_b} \int_{\mathcal{B}_i} \mathbf{b}_i \cdot \mathbf{v} \, dV + \sum_{j=1}^{N_t} \int_{\delta\mathcal{B}_j} \mathbf{t}_j \cdot \mathbf{v} \, dA + \sum_{k=1}^{N_f} \mathbf{f}_k \cdot \mathbf{v}_k$$

is the power of external loads over \mathcal{B}_t and its boundary.

Virtual work: the application of the D'Alembert Principle of classical mechanics gives the weak form of the mechanical equilibrium. Using virtual displacements as test functions, the variation of the mechanical energy potentials will vanish for any compatible variation:

$$\delta\Pi = \frac{\partial\Pi}{\partial x_i} \delta x_i = 0 \quad (4.8)$$

$$\delta\Pi_{mech} = 0 = \delta\Pi_{inertia} - \delta\Pi_{external} \quad (4.9)$$

where each of the previous contribution terms can be expressed as:

$$\begin{aligned} \delta\Pi_{inertia} &= \sum_{i=1}^{N_d} \int_{\mathcal{B}_i} \rho_i \dot{\mathbf{v}} \cdot \delta \mathbf{x} \, dV + \sum_{j=1}^{N_p} m_j \dot{\mathbf{v}}_j \cdot \delta \mathbf{x}_j \\ \delta\Pi_{external} &= \sum_{i=1}^{N_b} \int_{\mathcal{B}_i} \mathbf{b}_i \cdot \delta \mathbf{x} \, dV + \sum_{j=1}^{N_t} \int_{\delta\mathcal{B}_j} \mathbf{t}_j \cdot \delta \mathbf{x} \, dA + \sum_{k=1}^{N_f} \mathbf{f}_k \cdot \delta \mathbf{x}_k \end{aligned}$$

4.2 Basic points coordinates

The movement of a rigid body embedded in an Euclidean Space of dimension n , \mathbb{E}^n , can be described exactly by the spatial position of a minimum of n points. Taking into account that the number of parameters needed for establishing the position of those n points is n^2 (unknowns) and that the enforcing of the constant distance conditions of n points implies the same number of conditions (equations) for each additional point, the number of independent unknowns of the problem of the parameterization of a rigid body or *degrees of freedom* (DOF) are given by the next expression:

$$\text{DOF} = n^2 - \sum_{i=1}^n (i-1) \quad (4.10)$$

For a system composed of r -rigid bodies, with a set of joints between them that apply a total of s -constraints of relative displacement, the previous statement can be generalized:

$$\text{Mechanical system's DOF} = r \cdot \left[n^2 - \sum_{i=1}^n (i-1) \right] - s \quad (4.11)$$

The selection of the points and associated coordinates used to describe the con-

figuration of the body define the type of parameterization:

Independent coordinates: are the ones that use a number of parameters equal to the degrees of freedom of the rigid multi-body system. Its use is not frequent as it requires a process of selection of parameters that is very difficult to systematize. Also, there can be situations in which the configuration of some systems cannot be uniquely defined using this kind of parameterization.

Relative coordinates: This kind of discretization relies in the relative degrees of freedom between the bodies to define its configuration. Each one of the parameters directly defines the magnitude of one degree of freedom associated to a mechanical joint. The relative coordinates represent the minimum set of parameters needed to define an open chain mechanism. In the case where there are closed chains, a—sometimes complex—preprocessing for substituting them for additional constraints is needed.

Reference point coordinates: The position of each of the bodies that compose the system is defined through the position and orientation of a reference frame fixed to one of its points, usually the center of gravity. The number of cartesian and angular parameters usually corresponds exactly to the degrees of freedom of the body so no internal constraints must be added. The advantage of this parameterization is that its implementation is easy to generalize for any body, although it requires a slightly higher number of unknowns and equations, as each of the constraints between the bodies must be explicitly imposed.

Natural coordinates: Instead of selecting an arbitrary point of a body, the points where the joints are located are chosen [20]. If at least two points for each body are used, there is no need to use angular parameters, reducing the complexity of the formulation specially in 3D dynamics. Each point is described by its cartesian coordinates and each body is defined by the position of different points and, sometimes, also by some orientation vectors, leading to a constant mass matrix [22]. As each of the points located in the joints are used for describing the position of the bodies joined, the number of total parameters for a system is reduced.

Basic points coordinates represent an intermediate approach of the later two parameterizations. The reference frame is located in the center of gravity, the points are selected in the principal inertia directions forming an orthonormal basis, and only cartesian coordinates are used to avoid the use of angular parameters. The reference frame used to defined a body is formed by a set of points with an associated mass matrix. Later on it will be shown that this

presents a great advantage as the mass matrix will be diagonal and also remain constant during the movement. A set of constraints must be imposed between the points, increasing the number of degrees of freedom and equations needed. But, as the multi-body systems will also have flexible bodies with a much higher number of parameters, this increment of the variables and equations is less important than in the case of considering only rigid bodies.

Independently of the discretization used in the formulation, the position of any point of a rigid body will be defined at any time using a finite set of parameters called *generalized coordinates*, \mathbf{q}_R :

$$\mathbf{x} = \mathbf{x}(\mathbf{q}_R), \quad (4.12)$$

and, in the case of using natural or reference points coordinates, the relation between the position of any point and the generalized coordinates is constant and given by its transformation matrix:

$$\mathbf{x}_p = \mathbf{C}_p \cdot \mathbf{q}_R \quad (4.13)$$

Taking into account that a virtual displacement of any point of the rigid body is $\delta \mathbf{x} = \frac{\partial \mathbf{x}}{\partial \mathbf{q}_R} \delta \mathbf{q}_R$, the discretized weak form is given by:

$$\delta \mathbf{q}_R^T \cdot (\mathbf{M}_r \ddot{\mathbf{q}}_R - \mathbf{Q}_R) = 0 \quad , \quad \forall \delta \mathbf{q}_r \text{ constraint compatible} , \quad (4.14)$$

\mathbf{M}_r being the mass matrix and $\mathbf{Q}_r = \mathbf{Q}_g + \mathbf{Q}_f$ the generalized force vector. The term \mathbf{Q}_g represents the gyroscopic-type forces associated to the selected parametrization, and \mathbf{Q}_f the forces related to the loads applied to the body. Further elaboration of the different terms of (4.14) for any kind of parameterization is out of the scope of this work and can be found in several references [21, 66, 2].

Obviously, an independent set of coordinates would allow us to cancel the term inside the parenthesis in (4.14), resulting in a set of n ordinary differential equations describing the rigid body motion. This is not possible for a dependent set of coordinates, which will require the addition of constraint equations and the introduction of additional terms related with the constraint forces.

For a mechanism composed exclusively of rigid bodies, the different terms in (4.14) can be further elaborated as follows. Assuming that the rigid body is composed by N_p discrete points with mass and N_d rigid parts with a distributed mass, \mathbf{M}_R is the mass matrix, given by:

$$M_{R_{ij}} = \sum_{k=1}^{N_p} m_k \frac{\partial \mathbf{x}_k}{\partial q_{R_i}} \frac{\partial \mathbf{x}_k}{\partial q_{R_j}} + \sum_{l=1}^{N_d} \left[\int_{B_j} \rho_l \frac{\partial \mathbf{x}}{\partial q_{R_i}} \frac{\partial \mathbf{x}}{\partial q_{R_j}} dV \right] \quad (4.15)$$

What is more, the generalized force vector \mathbf{Q}_R takes the form $\mathbf{Q}_R = \mathbf{Q}_g + \mathbf{Q}_f$. The term \mathbf{Q}_g represents the gyroscopic-type forces associated to the selected parametrization, and \mathbf{Q}_f the forces related to the loads applied to the body, given by:

$$\mathbf{Q}_{f_i} = \sum_{j=1}^{N_f} f_j \frac{\partial \mathbf{x}_j}{\partial q_{R_i}} + \sum_{k=1}^{N_t} \left[\int_{\delta B_k} t_k \frac{\partial \mathbf{x}}{\partial q_{R_i}} dA \right] + \sum_{l=1}^{N_b} \left[\int_{B_l} b_l \frac{\partial \mathbf{x}}{\partial q_{R_i}} dV \right] \quad (4.16)$$

assuming the cartesianly decomposed N_f concentrated forces f_j , N_t surface loads t_k and N_b volumetric loads b_l .

The particular choice of the basic points coordinates results in a dependent set of coordinates, requiring the satisfaction of holonomic constraints related with the selected points. This approach is very systematic and possesses several additional advantages as it eliminates the gyroscopic-type forces in the equations of motion and produces a constant mass matrix.

Based on the previous results, the weak formulation of a system of rigid bodies connected by joints is straightforward. Collecting the coordinates of all rigid bodies under vector \mathbf{q}_R , the equations of motion are formally obtained replacing \mathbf{q}_r by \mathbf{q}_R in (4.14). But note that in this case $\delta \mathbf{q}_R$ are never compatible, even if the set of coordinates for each single body is independent, because the constraints introduced by the joints.

4.3 Formulation of rigid bodies

Once the type of discretization has been defined, any of the points of a rigid body can be located during its movement. Any rigid body will be described by a set of points and constraints. The inertial components will be studied in this section while the internal forces, as it is produced by the constant distance constraints, will be analyzed in Section 4.4.

The contributions to the *global mass matrix* can be proposed for each of the bodies, classified depending on their dimension, i.e. the space where their mass is distributed. The most simple body is a point mass, considered a 0-D element as it ideally does not occupy any space. Considering a system moving in a n -dimension space and a value of the point mass of m_p , the local mass matrix is as simple as:

$$\mathbf{M}_p = m_p \mathbf{I}_n \quad (4.17)$$

For bodies with distributed masses, the formulation is more involved and will be detailed in the following subsections.

4.3.1 1-D bodies: Bars

A bar is a rigid body with its mass, m_b , distributed along a line. The inertia around that line is zero, so it can be modeled using only two aligned points. If the end points, A, B of a bar of length l_b are chosen for its discretization, the position of any other point belonging to the bar is defined by one parameter in the local coordinate frame as:

$$\mathbf{x} = \mathbf{x}_A + c_1 (\mathbf{x}_B - \mathbf{x}_A), [0 \leq c_1 \leq 1] \quad (4.18)$$

which can be put in matrix notation:

$$\mathbf{x} = \begin{bmatrix} (1 - c_1)\mathbf{I}_n & c_1\mathbf{I}_n \end{bmatrix} \begin{Bmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{Bmatrix} = \mathcal{C} \cdot \mathbf{q} \quad (4.19)$$

The constant matrix \mathcal{C} represents the relation between the parameterization and coordinates of any point of the bar, $\frac{\partial \mathbf{x}_i}{\partial q_j} = C_{ij}$. The mass matrix can be calculated using the definition of the of Eq. 4.15:

$$\mathbf{M}_R = \int_{B_b} \rho_b \mathcal{C}^T \mathcal{C} dV \quad (4.20)$$

with:

$$\mathcal{C}^T \mathcal{C} = \begin{bmatrix} (1 - c_1)^2 \mathbf{I}_n & (1 - c_1)c_1 \mathbf{I}_n \\ (1 - c_1)c_1 \mathbf{I}_n & c_1^2 \mathbf{I}_n \end{bmatrix} \quad (4.21)$$

$$\rho_b(c_1) = \frac{m_b}{\int_{l_b} dl} = \frac{m_b}{[c_1]_0^1} = m_b \quad (4.22)$$

where the density has been considered constant along the bar. The substitution and integration leads to the following mass matrix of a rigid bar:

$$\mathbf{M}_R = \begin{bmatrix} \frac{m_b}{3} \mathbf{I}_n & \frac{m_b}{6} \mathbf{I}_n \\ \frac{m_b}{6} \mathbf{I}_n & \frac{m_b}{3} \mathbf{I}_n \end{bmatrix} \quad (4.23)$$

If the center of mass, \mathbf{x}_0 and director vector $\mathbf{d}_1 = \mathbf{x}_1 - \mathbf{x}_0$ are used as basic points of the parameterization, any point of the body can be expressed as:

$$\mathbf{x} = \mathbf{x}_0 + c_1 \mathbf{d}_1, \left[-\frac{l}{2} \leq c_1 \leq \frac{l}{2} \right] \quad (4.24)$$

the constant matrix of the body will have the next simple form:

$$\mathcal{C} = \begin{bmatrix} \mathbf{I}_n & c_1 \mathbf{I}_n \end{bmatrix} \quad (4.25)$$

and the mass matrix can be calculated as previously:

$$\mathbf{M}_R = \int_{-\frac{l}{2}}^{\frac{l}{2}} \rho_b \begin{bmatrix} \mathbf{I}_n & c_1 \mathbf{I}_n \\ c_1 \mathbf{I}_n & c_1^2 \mathbf{I}_n \end{bmatrix} dl = \begin{bmatrix} m_b \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & I_o \mathbf{I}_n \end{bmatrix} \quad (4.26)$$

where:

$I_o = \frac{\rho_b l^3}{12}$ is the polar moment of inertia of the bar around a perpendicular axis passing through its center of mass.

Note that body's constant matrix is built by a function (i.e. 4.24) that represents the relation between any body point with the nodal coordinates. In essence, it represents the same transformation as the shape functions provide for flexible bodies. What's more, the polynomial equation 4.19 is equivalent to the shape function of a 1D element when described in the real space (as opposed to the isoparametric definition).

4.3.2 2-D bodies: Plates

A plate is a solid that has all of its points contained in a plane. The 2-axis reference frame needed to define the position of any point of the plate can be formed using any three non-aligned points. The center of gravity, \mathbf{x}_0 , and two other points, $\{\mathbf{x}_1, \mathbf{x}_2\}$ are selected such that the director vectors, $\mathbf{d}_1 = (\mathbf{x}_1 - \mathbf{x}_0)$, and, $\mathbf{d}_2 = (\mathbf{x}_2 - \mathbf{x}_0)$ define an orthonormal reference system, coincident with the principal axis of the solid.

The position of any point of the body can be expressed as:

$$\mathbf{x} = \mathbf{x}_0 + c_1 \mathbf{d}_1 + c_2 \mathbf{d}_2 \quad (4.27)$$

the constant matrix of the body will have the next simple form:

$$\mathcal{C} = \begin{bmatrix} 1 \mathbf{I}_n & c_1 \mathbf{I}_n & c_2 \mathbf{I}_n \end{bmatrix} \quad (4.28)$$

and the mass matrix can be calculated as previously:

$$\mathbf{M}_R = \int_B \rho_p \begin{bmatrix} \mathbf{I}_n & c_1 \mathbf{I}_n & c_2 \mathbf{I}_n \\ c_1 \mathbf{I}_n & c_1^2 \mathbf{I}_n & c_1 c_2 \mathbf{I}_n \\ c_2 \mathbf{I}_n & c_1 c_2 \mathbf{I}_n & c_2^2 \mathbf{I}_n \end{bmatrix} dA = \begin{bmatrix} m_p \mathbf{I}_n & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_1 \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_2 \mathbf{I}_n \end{bmatrix} \quad (4.29)$$

4.3.3 3-D bodies

A general 3D body is modeled using four points, $\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, and six constraints for maintaining the orthonormal triad that they form. The local reference frame is located in the center of gravity, \mathbf{x}_0 and its director vectors are defined using

the other three points, $\mathbf{d}_i = \mathbf{x}_i - \mathbf{x}_0$.

Using this local reference frame, the position of a material point of the body is calculated by:

$$\mathbf{x} = \mathbf{x}_0 + c_1 \mathbf{d}_1 + c_2 \mathbf{d}_2 + c_3 \mathbf{d}_3 \quad (4.30)$$

$$\mathcal{C} = \begin{bmatrix} \mathbf{I}_3 & c_1 \mathbf{I}_3 & c_2 \mathbf{I}_3 & c_3 \mathbf{I}_3 \end{bmatrix} \quad (4.31)$$

$$\begin{aligned} \mathbf{M}_R &= \int_B \rho_p \begin{bmatrix} \mathbf{I}_3 & c_1 \mathbf{I}_3 & c_2 \mathbf{I}_3 & c_3 \mathbf{I}_3 \\ c_1 \mathbf{I}_3 & c_1^2 \mathbf{I}_3 & c_1 c_2 \mathbf{I}_3 & c_1 c_3 \mathbf{I}_3 \\ c_2 \mathbf{I}_3 & c_1 c_2 \mathbf{I}_3 & c_2^2 \mathbf{I}_3 & c_2 c_3 \mathbf{I}_3 \\ c_3 \mathbf{I}_3 & c_1 c_3 \mathbf{I}_3 & c_2 c_3 \mathbf{I}_3 & c_3^2 \mathbf{I}_3 \end{bmatrix} dA \\ &= \begin{bmatrix} m_p \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_1 \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_2 \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I_3 \mathbf{I}_3 \end{bmatrix} \end{aligned}$$

4.4 Generalized joints formulation

The constraints relate the position or orientation of material points. In rigid bodies, constraints are imposed between different bodies to model joints. As it will be shown in section 4.3, constraint equations are needed when the rigid body is modelled with more nodes than necessary, so they do not have always a physical counterpart as it is always the case with joints. If the bodies are flexible, internal constraints can be also added to model an specific behaviour (e.g. a reinforcement) as it was detailed in section 3.2.1.

This section extends and details the different techniques for applying constraints that were introduced for flexible discrete bodies. Some typical constraint equations in solid bodies (rigid and flexible) are described, and combined to form joints. The application of joints in rigid bodies is summarized along with a generic way to impose the same formulae to joints including flexible bodies. The methodology is based in the application of meshfree shape functions for the definition of the constraint points. This *generalized* form is applicable to any joint connecting any kind of rigid or flexible body. It's also independent on the discretization method used for the flexible bodies, therefore being compatible with FEM or meshfree shape functions.

4.4.1 Application of joints to flexible bodies

From the practical point of view, note that the use of a meshfree discretization greatly simplifies the modelling of joints, because nodes can be added at any place inside the boundary, independently of the integration grid. In a computer combined design/analysis process of a real mechanism, the user may introduce corrections on the design very efficiently from the computational point of view, since no connectivity must be rebuilt, and therefore any added/removed joint has small impact in the overall data structure of the software.

The main issue regarding the modelling of joints connecting deformable bodies discretized with a meshfree technique is related to the delta property of the shape functions. This property states that the value of the shape function associated to a given support node is 1 at the node, and 0 at the rest of them, and it is a basic property of FEM. This property greatly simplifies the formulation of joints, because a constraint applied on specific nodes of the flexible body just relates the degrees of freedom of the affected nodes. For some simple cases (e.g. constraints that fix the position of nodes, or constraints over coincident nodes of different bodies, sharing several degrees of freedom), the constraint can be enforced just eliminating the affected degrees of freedom. Note also that the delta property must be satisfied when dealing with continuous varying boundary conditions for achieving at least linear accuracy (*patch test*).

Not all meshfree techniques have this delta property; actually some of the most common types of shape functions lack this property, as MLS and RKPM. When it does not hold, the formulation of constraints is more involved; direct elimination is not possible for simple constraints, requiring the addition of constraint equations to couple the desired degrees of freedom, even if the nodes are coincident. For these cases, some techniques can be developed to recover this property, as the *transformation method* [15] or the *coupling with finite elements* [42].

However, some meshfree shape functions do have the delta property. This is the case of the RBF interpolants, and this is actually one of the main reasons for selecting them to be used in a multibody context, where an efficient representation of joints plays such an important role in the overall formulation. With these RBF interpolants, joints can be modelled as simple as with the use of FEM, eliminating degrees of freedom for simple joints and facilitating the general formulation of more complex constraints that have to be imposed explicitly.

The following presents a methodology for connecting flexible bodies between them or to rigid bodies. As opposed to the common approaches, it is applicable to any meshfree methods, having or not the Kronecker delta property. Differences between its application to both types of methods will be commented, specially in

what regards to efficiency and errors in the constraint.

4.4.2 Mechanical constraint equations

As pointed out in the previous section, kinematic joints among the different parts of the mechanism are represented by a set of constraint equations. We will focus in *holonomic constraints*, which are those that do not depend on the derivatives of the generalized coordinates

$$\Phi(q) = 0 \quad (4.32)$$

The classical methods to introduce their effect in the equations of motion are Lagrange multipliers, penalty and the augmented Lagrange method. A more detailed description of these methods is given in the following paragraphs.

The Lagrange multipliers add the following term to the general dynamic equation 4.14:

$$\delta q^T \{M\ddot{q} - Q\} + \delta(\lambda^T \Phi) = 0$$

resulting the system:

$$\delta q^T \{M\ddot{q} + (\Phi_q^T \lambda) - Q\} + \delta \lambda^T \Phi = 0$$

where the index indicates the first derivative: $\Phi_q = \frac{\partial \Phi}{\partial q}$

The previous equation holds for any variation $\delta q, \delta \lambda$, so both addition terms must vanish. The system is decomposed in two complementary systems of n second order differential equations and p algebraic equations, forming the next DAE system:

$$M\ddot{q} + (\Phi_q^T \lambda) - Q = 0 \quad (4.33)$$

$$\Phi = 0 \quad (4.34)$$

To facilitate the numerical resolution of the DAE system, the constraints can be substituted by their derivatives. The derivation of the expression $\Phi = 0$ twice leads to:

$$\dot{\Phi} = 0 \rightarrow \Phi_q \dot{q} + \Phi_t = 0 \quad (4.35)$$

$$\ddot{\Phi} = 0 \rightarrow \Phi_q \ddot{q} + \frac{d\Phi_q}{dt} \dot{q} + \frac{d\Phi_t}{dt} = 0 \quad (4.36)$$

Swaping Φ by its second derivative and grouping the terms of the Eq. (4.33),

the final form of the dynamic system is obtained:

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_q^T \\ \mathbf{\Phi}_q & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathcal{Q} \\ \mathbf{g} \end{Bmatrix} \quad (4.37)$$

where:

$$\mathbf{g} = -\frac{d\mathbf{\Phi}_q}{dt} \dot{\mathbf{q}} - \frac{d\mathbf{\Phi}_t}{dt}$$

The application of the penalty method substitutes the multipliers, λ , in Eq. (4.33) by a product of the constraint equations and a *penalty matrix*, $\alpha = \alpha_{ii}$:

$$\lambda \simeq \alpha \cdot \Phi \quad (4.38)$$

The resulting second order ODE system will then maintain its original size:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_q^T (\alpha \cdot \Phi) - \mathcal{Q} = \mathbf{0} \quad (4.39)$$

As the penalization parameters' values are increased infinitely, $\alpha_{ii} \rightarrow \infty$, the constraints are more exactly imposed, implying that $(\alpha \cdot \Phi) \rightarrow \lambda$, although the system becomes more numerically ill-conditioned.

The augmented lagrange method represents an intermediate approach of the previous two methods. Making the following substitution in Eq. (4.33):

$$\lambda = \alpha \cdot \Phi + \lambda^* \quad (4.40)$$

the system is transformed to the following ODE system:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_q^T (\alpha \cdot \Phi) + \mathbf{\Phi}_q^T \lambda^* - \mathcal{Q} = \mathbf{0} \quad (4.41)$$

Where the vector λ^* can be considered a correction to the penalty forces by means of Lagrange multipliers.

In the Eq. (4.41) is obvious that $\mathbf{\Phi}_q^T (\alpha \cdot \Phi) \rightarrow \mathbf{0}$ if and only if $\lambda^* \rightarrow \lambda$. If an iterative method in λ^* is used, so it converges to λ the system constraints will be imposed exactly.

Retaking the idea of the penalty method, where the constraint is imposed with more precision as α increases and noting that the constraint is imposed exactly using the Lagrange multipliers, it can be stated that the next succession is monotonic and converges to λ :

$$\lambda_{i+1}^* = \lambda_i^* + \alpha \cdot \Phi_{i+1} \quad (4.42)$$

If $\Phi_{i+1} = \Phi(\mathbf{q}_i)$ is calculated with the solution for the displacements given by

Eq. (4.41), then:

$$\mathbf{M}\ddot{\mathbf{q}}_i + (\Phi_q^T)_i (\alpha \cdot \Phi_i) + (\Phi_q^T)_i \lambda_i^* - Q_i = 0 \quad (4.43)$$

A. Constant distance between two points

In the reference configuration, the distance between two points of different bodies with positions \mathbf{X}_1 and \mathbf{X}_2 is defined as:

$$d_0 = \|\mathbf{X}_2 - \mathbf{X}_1\| = \sqrt{\sum_{i=1}^{dim} (X_{2i} - X_{1i})^2} \quad (4.44)$$

With dim the dimension space of the problem.

At any time, the distance between both points will have a magnitude d_t :

$$d_t = \|\mathbf{x}_2 - \mathbf{x}_1\| = \sqrt{\sum_{i=1}^{dim} (x_{2i}(t) - x_{1i}(t))^2} \quad (4.45)$$

The constraint imposes the distance to remain equal as originally, and is usually applied as a squared exponential to reduce the number of calculations as the previous squared roots would not be required:

$$\Phi_i = \Phi_i = d_t^2 - d_0^2 \quad (4.46)$$

Another advantage appears when computing the derivatives of the constraint, as the expressions get greatly simplified. For example, in the case of a 2-dimensional problem, the first derivative results in the following:

$$(\Phi_q)_i = \nabla_q \Phi_i = \nabla d_t^2 = \begin{pmatrix} -2 (x_2(t) - x_1(t)) \\ -2 (y_2(t) - y_1(t)) \\ 2 (x_2(t) - x_1(t)) \\ 2 (y_2(t) - y_1(t)) \end{pmatrix}^T \quad (4.47)$$

When one of the points belongs to a rigid body, its position \mathbf{x} is calculated using the constant matrix \mathcal{C} , as defined e.g. in equation (4.19):

$$\mathbf{x} = \mathcal{C} \cdot \mathbf{q} \quad (4.48)$$

Or, for each of the $i = (x, y, z)$ components:

$$x_i = \sum_{j=1}^{dim+1} \mathcal{C}_{ij} q_j \quad (4.49)$$

Without loss of generality, considering that both constrained points are part of

a pair of similar (i.e. same dimension) rigid bodies, A and B . Let's assume that the bodies are a couple of 1D rigid bars, then the constraint can be formulated as:

$$\Phi_i = \Phi_i = d_t^2 - d_0^2 \quad (4.50)$$

$$= \sum_{i=1}^{dim} (x_{2i}(t) - x_{1i}(t))^2 - d_0^2 \quad (4.51)$$

$$= \sum_{i=1}^{dim} \left(\sum_{j=1}^{dim+1} (c_{ij}^B q_j^B - c_{ij}^A q_j^A) \right)^2 - d_0^2 \quad (4.52)$$

Returning to the example of the constraint in 2D space, the first derivative can then be formulated also as a function of the rigid body nodes generalized coordinates. For two bars, each point will have two support nodes, and each node, two degrees of freedom. The total size of the derivative corresponds to the eight generalized coordinates of the joint:

$$(\Phi_q)_i = \nabla_q \Phi_i = \nabla d_t^2 = \begin{pmatrix} -2 d_t \\ -2 d_t \\ -2 d_t \\ -2 d_t \\ 2 d_t \\ 2 d_t \\ 2 d_t \\ 2 d_t \end{pmatrix}^T \quad (4.53)$$

Let's note that using meshfree shape functions, the position of any point inside or outside the domain can be calculated from the support nodes' (SP) location (as stated by eq. 3.1):

$$x_i(\mathbf{X}, t) \simeq x_i^h(\mathbf{X}, t) = \sum_{j=1}^{SP} N_j(\mathbf{X}) q_{ij}(t) \quad (4.54)$$

with $i = \{1, \dots, m\text{-DOFs}\}$

Applying the same procedure as for the rigid body constant matrix, the constraint can be imposed between points belonging to a pair of flexible bodies A and B . The support nodes for each node will be different in general, and here are denoted

as $SP1$ and $SP2$ respectively. The constraint has the following form:

$$\Phi_i = \Phi_i = \sum_{i=1}^{dim} (x_{2i}(t) - x_{1i}(t)) - d_0^2 \quad (4.55)$$

$$= \sum_{i=1}^{dim} \left(\sum_{j=1}^{SP2} N_j^B(\mathbf{X}^B) q_{ij}(t) - \sum_{k=1}^{SP1} N_k^A(\mathbf{X}^A) q_{ik}(t) \right)^2 - d_0^2 \quad (4.56)$$

The previous expression is valid for any kind of discretization of the flexible body. But, if the shape functions have the Kronecker delta property and the point selected to impose the constraint is a node, there is no need to compute any sum. In this particular case, the displacement of that node is directly its generalized coordinate, speeding up the calculation and eliminating any approximation error.

Finally, going back to the 2D case between two flexible bodies, the expression for the derivative gets a generic form. This resulting derivative of the constraint has a general form of a transposed vector of size $(dim(SP1 + SP2))$, being valid for any dimension and type of solid:

$$(\Phi_q)_i = \nabla_q \Phi_i = \nabla d_t^2 = \begin{pmatrix} -2 d_t \\ -2 d_t \\ \vdots \\ -2 d_t \\ 2 d_t \\ 2 d_t \\ \vdots \\ 2 d_t \end{pmatrix}^T \quad (4.57)$$

B. Fixed coordinates between points

When the points are in the same location, an alternative and even more direct formulation is defined using only each of the coordinates:

$$\Phi_i = \begin{pmatrix} \Phi_{ix} \\ \Phi_{iy} \\ \Phi_{iz} \end{pmatrix} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix} \quad (4.58)$$

The previous expression can be written as a function of the supporting nodes using numerical indexing of coordinates $\{x, y, z\} = \{1, 2, 3\}$, in a similar manner as before:

$$\Phi_i = \begin{pmatrix} \sum_{j=1}^{SP2} N_j^B(\mathbf{X}^B) q_{1j}(t) - \sum_{k=1}^{SP1} N_k^A(\mathbf{X}^A) q_{1k}(t) \\ \sum_{j=1}^{SP2} N_j^B(\mathbf{X}^B) q_{2j}(t) - \sum_{k=1}^{SP1} N_k^A(\mathbf{X}^A) q_{2k}(t) \\ \sum_{j=1}^{SP2} N_j^B(\mathbf{X}^B) q_{3j}(t) - \sum_{k=1}^{SP1} N_k^A(\mathbf{X}^A) q_{3k}(t) \end{pmatrix} \quad (4.59)$$

Which can be simplified to the following:

$$\Phi_i = \sum_{j=1}^{SP2} N_j^B(\mathbf{X}^B) \begin{pmatrix} q_{1j}(t) \\ q_{2j}(t) \\ q_{3j}(t) \end{pmatrix} - \sum_{k=1}^{SP1} N_k^A(\mathbf{X}^A) \begin{pmatrix} q_{1k}(t) \\ q_{2k}(t) \\ q_{3k}(t) \end{pmatrix} \quad (4.60)$$

The derivatives are then computed in a generalized way leading to the following Jacobian matrix, with a size of $(dim \times dim(SP1 + SP2))$:

$$(\Phi_q)_i = \nabla_q \Phi_i = \begin{pmatrix} -N_1^A(\mathbf{X}^A) \mathbf{I}_{dim} \\ -N_2^A(\mathbf{X}^A) \mathbf{I}_{dim} \\ \vdots \\ -N_{SP1}^A(\mathbf{X}^A) \mathbf{I}_{dim} \\ N_1^B(\mathbf{X}^B) \mathbf{I}_{dim} \\ N_2^B(\mathbf{X}^B) \mathbf{I}_{dim} \\ \vdots \\ N_{SP2}^B(\mathbf{X}^B) \mathbf{I}_{dim} \end{pmatrix}^T \quad (4.61)$$

Note that when the delta property holds, the constraint can be imposed using eq. 4.58, and its derivative is greatly simplified:

$$(\Phi_q)_i = \nabla_q \Phi_i = \begin{pmatrix} -\mathbf{I}_{dim} \\ \mathbf{I}_{dim} \end{pmatrix}^T \quad (4.62)$$

C. Alignment of three points in space

Any three points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are aligned if the cross product between their relative vectors is zero. Two of them are used for defining the constraint: $\mathbf{P}_1 = \mathbf{x}_2 - \mathbf{x}_1$, and $\mathbf{P}_2 = \mathbf{x}_3 - \mathbf{x}_1$. The alignment condition enforces the following product to vanish:

$$\Phi_i = \begin{pmatrix} \Phi_{ix} \\ \Phi_{iy} \\ \Phi_{iz} \end{pmatrix} = \mathbf{P}_1 \times \mathbf{P}_2 \quad (4.63)$$

The vectors can be defined as a function of the generalized coordinates as in eq. 4.60 using the point support sums:

$$\mathbf{P}_1 = \sum_{i=1}^{SP2} N_j(\mathbf{X}^2) \begin{pmatrix} q_{1i}(t) \\ q_{2i}(t) \\ q_{3i}(t) \end{pmatrix} - \sum_{k=1}^{SP1} N_k(\mathbf{X}^1) \begin{pmatrix} q_{1k}(t) \\ q_{2k}(t) \\ q_{3k}(t) \end{pmatrix} \quad (4.64)$$

$$\mathbf{P}_2 = \sum_{j=1}^{SP3} N_j(\mathbf{X}^3) \begin{pmatrix} q_{1j}(t) \\ q_{2j}(t) \\ q_{3j}(t) \end{pmatrix} - \sum_{i=1}^{SP2} N_j(\mathbf{X}^2) \begin{pmatrix} q_{1i}(t) \\ q_{2i}(t) \\ q_{3i}(t) \end{pmatrix} \quad (4.65)$$

Let's define the third vector relative to the points as $\mathbf{P}_3 = \mathbf{x}_3 - \mathbf{x}_2$

$$\mathbf{P}_3 = \sum_{j=1}^{SP3} N_j(\mathbf{X}^3) \begin{pmatrix} q_{1j}(t) \\ q_{2j}(t) \\ q_{3j}(t) \end{pmatrix} - \sum_{k=1}^{SP1} N_k(\mathbf{X}^1) \begin{pmatrix} q_{1k}(t) \\ q_{2k}(t) \\ q_{3k}(t) \end{pmatrix} \quad (4.66)$$

After some work with the equations, the derivative in 3D space can be expressed in the following form:

$$\left(\Phi_q \right)_i = \nabla_q \Phi_i = \left(\Phi_{i1} \quad \Phi_{i2} \quad \Phi_{i3} \right) \quad (4.67)$$

With each of the sub-matrices having this transposed form:

$$\Phi_{i1} = \begin{pmatrix} N_1(\mathbf{X}^1) \mathbf{A} \\ N_2(\mathbf{X}^1) \mathbf{A} \\ \vdots \\ N_{SP1}(\mathbf{X}^1) \mathbf{A} \end{pmatrix}^T \quad (4.68)$$

$$\Phi_{i2} = \begin{pmatrix} N_1(\mathbf{X}^2) \mathbf{B} \\ N_2(\mathbf{X}^2) \mathbf{B} \\ \vdots \\ N_{SP2}(\mathbf{X}^2) \mathbf{B} \end{pmatrix}^T \quad (4.69)$$

$$\Phi_{i3} = \begin{pmatrix} N_1(\mathbf{X}^3) \mathbf{C} \\ N_2(\mathbf{X}^3) \mathbf{C} \\ \vdots \\ N_{SP3}(\mathbf{X}^3) \mathbf{C} \end{pmatrix}^T \quad (4.70)$$

With the skew-symmetric matrices:

$$\mathbf{A} = \begin{pmatrix} 0 & P_{3z} & -P_{3y} \\ -P_{3z} & 0 & P_{3x} \\ P_{3y} & -P_{3x} & 0 \end{pmatrix} \quad (4.71)$$

$$\mathbf{B} = \begin{pmatrix} 0 & -P_{2z} & P_{2y} \\ P_{2z} & 0 & -P_{2x} \\ -P_{2y} & P_{2x} & 0 \end{pmatrix} \quad (4.72)$$

$$\mathbf{C} = \begin{pmatrix} 0 & P_{1z} & -P_{1y} \\ -P_{1z} & 0 & P_{1x} \\ P_{1y} & -P_{1x} & 0 \end{pmatrix} \quad (4.73)$$

D. Constant orientation between two vectors

Two vectors can be defined in a pair of bodies by a set of four points, each two defining the corresponding vector of interest in the body i : $(\mathbf{P}_i)_0 = \mathbf{X}_{12}^i = \mathbf{X}_2^i - \mathbf{X}_1^i$. The evolution of each vector during the deformation of each body will then be $(\mathbf{P}_i)_t = \mathbf{x}_{12}^i = \mathbf{x}_2^i - \mathbf{x}_1^i$.

The initial angle between the two vectors can be calculated by the arcsin of the normalized cross product:

$$\theta_0 = \arcsin \frac{\|(\mathbf{P}_1)_0 \times (\mathbf{P}_2)_0\|}{\|(\mathbf{P}_1)_0\| \|(\mathbf{P}_2)_0\|} \quad (4.74)$$

Or also by the arccos of the normalized dot product

$$\theta_0 = \arccos \frac{\|(\mathbf{P}_1)_0 \cdot (\mathbf{P}_2)_0\|}{\|(\mathbf{P}_1)_0\| \|(\mathbf{P}_2)_0\|} \quad (4.75)$$

The constraint is defined by the differences between the angle at any time, θ_t , and the initial angle, θ_0 . To reduce the complexity of the derivatives, the trigonometric terms can be removed with some caution, as it must be taken into account that $\theta \simeq n\pi \rightarrow \sin(\theta) \simeq \theta$, while $\theta \simeq n\pi/2 \rightarrow \cos(\theta) \simeq \theta$, with $n = 0, 1, \dots$. Then the $\sin\theta$ along with equation 4.74 can be used for the ranges $[0 \leq |\theta_0| \leq \pi/4]$ and $[3\pi/4 \leq |\theta_0| \leq \pi]$; while the $\cos\theta$ with equation 4.75 would be used for the complementary angles $[\pi/4 \leq |\theta_0| \leq 3\pi/4]$.

$$\Phi_i = \theta_t - \theta_0 \simeq \begin{cases} \frac{\|(\mathbf{P}_1)_t \times (\mathbf{P}_2)_t\|}{\|(\mathbf{P}_1)_t\| \|(\mathbf{P}_2)_t\|} - \sin \theta_0, & 0 \leq |\theta_0| \leq \pi/4 \\ \frac{\|(\mathbf{P}_1)_t \cdot (\mathbf{P}_2)_t\|}{\|(\mathbf{P}_1)_t\| \|(\mathbf{P}_2)_t\|} - \cos \theta_0, & \pi/4 \leq |\theta_0| \leq 3\pi/4 \end{cases} \quad (4.76)$$

Considering the vector lengths as constant allows to simplify the formulation of the joint, which is valid if the points are selected adequately (i.e. in parts of both bodies with similar strains). This implies that $\|(\mathbf{P}_1)_t\| \|(\mathbf{P}_2)_t\| = \|(\mathbf{P}_1)_0\| \|(\mathbf{P}_2)_0\|$, and

the constraint is then:

$$\Phi_i = \begin{cases} \|(P_1)_t\| \times \|(P_2)_t\| - \|(P_1)_0\| \times \|(P_2)_0\|, & 0 \leq |\theta_0| \leq \pi/4 \\ \|(P_1)_t\| \cdot \|(P_2)_t\| - \|(P_1)_0\| \cdot \|(P_2)_0\|, & \pi/4 \leq |\theta_0| \leq 3\pi/4 \end{cases} \quad (4.77)$$

As in the case of three aligned points, the formulation of the derivatives when vector products appear are much more involved.

For the case of the first range of initial angles, $[\pi/4 \leq |\theta_0| \leq 3\pi/4]$, the jacobian results in the following:

$$(\Phi_q)_i = \nabla_q \Phi_i = (-\phi_{i1} \quad \phi_{i2} \quad -\phi_{i3} \quad \phi_{i4}) \quad (4.78)$$

With:

$$\phi_{i1} = \begin{pmatrix} -N_1(X^1)P_1 \\ -N_2(X^1)P_1 \\ \vdots \\ -N_{SP1}(X^1)P_1 \end{pmatrix}^T \quad (4.79)$$

$$\phi_{i2} = \begin{pmatrix} N_1(X^2)P_1 \\ N_2(X^2)P_1 \\ \vdots \\ N_{SP2}(X^2)P_1 \end{pmatrix}^T \quad (4.80)$$

$$\phi_{i3} = \begin{pmatrix} -N_1(X^3)P_2 \\ -N_2(X^3)P_2 \\ \vdots \\ -N_{SP3}(X^3)P_2 \end{pmatrix}^T \quad (4.81)$$

$$\phi_{i4} = \begin{pmatrix} -N_1(X^4)P_2 \\ -N_2(X^4)P_2 \\ \vdots \\ -N_{SP4}(X^4)P_2 \end{pmatrix}^T \quad (4.82)$$

And each of the vectors represented in generalized coordinates:

$$P_1 = \sum_{i=1}^{SP2} N_j(X^2) \begin{pmatrix} q_{1i}(t) \\ q_{2i}(t) \\ q_{3i}(t) \end{pmatrix} - \sum_{k=1}^{SP1} N_k(X^1) \begin{pmatrix} q_{1k}(t) \\ q_{2k}(t) \\ q_{3k}(t) \end{pmatrix} \quad (4.83)$$

$$\mathbf{P}_2 = \sum_{l=1}^{SP4} N_j(\mathbf{X}^4) \begin{pmatrix} q_{1l}(t) \\ q_{2l}(t) \\ q_{3l}(t) \end{pmatrix} - \sum_{j=1}^{SP3} N_j(\mathbf{X}^3) \begin{pmatrix} q_{1j}(t) \\ q_{2j}(t) \\ q_{3j}(t) \end{pmatrix} \quad (4.84)$$

Note that if the delta property holds, the derivative would be much more simple to calculate:

$$(\Phi_q)_i = \nabla_q \Phi_i = \begin{pmatrix} \mathbf{x}_3 - \mathbf{x}_4 \\ \mathbf{x}_4 - \mathbf{x}_3 \\ \mathbf{x}_1 - \mathbf{x}_2 \\ \mathbf{x}_2 - \mathbf{x}_1 \end{pmatrix}^T \quad (4.85)$$

In the other ranges of initial angles, $[0 \leq |\theta_0| \leq \pi/4]$ and $[3\pi/4 \leq |\theta_0| \leq \pi]$, the jacobian is much more involved and can be computed in a similar way as in the previous subsection, with the difference of the Euclidean norm.

4.4.3 Thermal constraint equations

The thermal constraints can be viewed as a simplification of the mechanical techniques, as they only affect to one degree of freedom and are applied to a first order discrete ODE system:

The Lagrange multipliers add the following term to the heat equation:

$$\delta \mathbf{q}^T \{ \mathbf{C} \dot{\mathbf{q}}_T - \mathbf{Q}^h \} + \delta (\lambda^T \Phi) = 0 \quad (4.86)$$

where, for simplification, \mathbf{Q}^h includes external heat and internal conduction. the system can be divided in:

$$\delta \mathbf{q}^T \{ \mathbf{C} \dot{\mathbf{q}}_T + (\Phi_{q_T}^T \lambda) - \mathbf{Q}^h \} + \delta \lambda^T \Phi = 0 \quad (4.87)$$

where the index indicates the first derivative with respect to the generalized temperatures: $\Phi_{q_T} = \frac{\partial \Phi}{\partial \mathbf{q}_T}$

The previous equation also holds for any variation $\delta \mathbf{q}, \delta \lambda$, so both addition terms must vanish. It can be rearranged to the next DAE system:

$$\mathbf{C} \dot{\mathbf{q}}_T + (\Phi_{q_T}^T \lambda) - \mathbf{Q}^h = \mathbf{0} \quad (4.88)$$

$$\Phi = \mathbf{0} \quad (4.89)$$

Instead of deriving twice as in the mechanical case, the constraints can be substituted by their first derivatives, leading to:

$$\dot{\Phi} = \mathbf{0} \rightarrow \Phi_{q_T} \dot{\mathbf{q}}_T + \Phi_t = \mathbf{0} \quad (4.90)$$

Swaping Φ by its first derivative and grouping the terms of the Eq. (4.88), the final form of the dynamic system is obtained:

$$\begin{bmatrix} \mathbf{C} & \Phi_{q_T}^T \\ \Phi_{q_T} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{q}_T \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{Q}^h \\ -\Phi_t \end{Bmatrix} \quad (4.91)$$

The application of the penalty method and Augmented Lagrange is equivalent to the process presented in section 4.4.2, leading to the following respective systems:

$$\mathbf{C} \dot{q} + \Phi_{q_T}^T (\alpha \cdot \Phi) - \mathbf{Q}^h = \mathbf{0} \quad (4.92)$$

$$\mathbf{C}(\dot{q})_{Ti} + (\Phi_{q_T}^T)_i (\alpha \cdot \Phi_i) + (\Phi_{q_T}^T)_i \lambda_i^* - \mathbf{Q}_i^h = \mathbf{0} \quad (4.93)$$

A. Fixed temperatures between points

When the points are in the same location, it is usual to link their temperatures. As these correspond to an additional degree of freedom, the same form from eq. 4.58 of the constraint for fixed coordinates can be used:

$$\Phi_i = \Theta_2 - \Theta_1 \quad (4.94)$$

Which, in generalized coordinates uses the support nodes for each point:

$$\Phi_i = \sum_{j=1}^{SP2} N_j^B(\mathbf{X}^B) q_{Tj}(t) - \sum_{k=1}^{SP1} N_k^A(\mathbf{X}^A) q_{Tk}(t) \quad (4.95)$$

And has a Jacobian matrix in form of a row vector, with size of $(SP1 + SP2)$:

$$(\Phi_{q_T})_i = \nabla_q \Phi_i = \begin{pmatrix} -N_1^A(\mathbf{X}^A) \\ -N_2^A(\mathbf{X}^A) \\ \vdots \\ -N_{SP1}^A(\mathbf{X}^A) \\ N_1^B(\mathbf{X}^B) \\ N_2^B(\mathbf{X}^B) \\ \vdots \\ N_{SP2}^B(\mathbf{X}^B) \end{pmatrix}^T \quad (4.96)$$

If the shape functions possess the Kronecker delta property, the derivative is as simple as:

$$(\Phi_q)_i = \nabla_q \Phi_i = \begin{pmatrix} -1 \\ 1 \end{pmatrix}^T \quad (4.97)$$

4.4.4 Joints between rigid bodies

A. 2D pin joint and 3D spherical joint

The joint is defined by a minimum of four points: 1,3 for one body, and 2,4 belonging to the other body. 1 and 2 are defined in the same initial location, setting the centre of rotation of the joint. The joint has therefore only rotational degree(s) of freedom. In 2D, the DoF is measured by the mean angle between the other non-coincident points of each body. In general, the coincident points will be defined by a set of support nodes in each body. Let there be N_A and N_B support nodes for the interfaces of the joint. A fixed point constraint can be imposed between the interfaces using eq 4.60:

$$\Phi_i = \sum_{j=1}^{SP2} N_j(\mathbf{X}^2) \begin{pmatrix} q_{1j}(t) \\ q_{2j}(t) \\ q_{3j}(t) \end{pmatrix} - \sum_{k=1}^{SP1} N_k(\mathbf{X}^1) \begin{pmatrix} q_{1k}(t) \\ q_{2k}(t) \\ q_{3k}(t) \end{pmatrix} \quad (4.98)$$

The joint orientation vectors are then defined in a similar way as:

$$\mathbf{P}_1 = \sum_{j=1}^{SP3} N_j(\mathbf{X}^3) \begin{pmatrix} q_{1j}(t) \\ q_{2j}(t) \\ q_{3j}(t) \end{pmatrix} - \sum_{k=1}^{SP1} N_k(\mathbf{X}^1) \begin{pmatrix} q_{1k}(t) \\ q_{2k}(t) \\ q_{3k}(t) \end{pmatrix} \quad (4.99)$$

$$\mathbf{P}_2 = \sum_{l=1}^{SP4} N_l(\mathbf{X}^4) \begin{pmatrix} q_{1l}(t) \\ q_{2l}(t) \\ q_{3l}(t) \end{pmatrix} - \sum_{j=1}^{SP3} N_j(\mathbf{X}^3) \begin{pmatrix} q_{1j}(t) \\ q_{2j}(t) \\ q_{3j}(t) \end{pmatrix} \quad (4.100)$$

And using a similar approach, the mean angle of rotation between the bodies can be calculated using equation 4.75, defined as the starting point for the vector alignment constraint:

$$\theta = \arccos \frac{\|\mathbf{P}_1 \cdot \mathbf{P}_2\|}{\|\mathbf{P}_1\| \|\mathbf{P}_2\|} \quad (4.101)$$

In the case of the 3D spherical joint, there are three degrees of freedom instead of just one, so another pair of points can be defined to measure the spatial angles, in order to have a complete definition of the relative positions of the bodies.

B. 3D revolute joint

A schematic representation of the joint is shown in 4.1. Six points are needed to define it. Five of them are used for setting the constraint equations:

- A constant distance constraint between two points defining the centre (as in the 2D case):

$$\Phi_i = \mathbf{x}_2 - \mathbf{x}_1 \quad (4.102)$$

- Two constant orientation constraints between a vector in the axis of rotation of the joint and two non-aligned vectors:

$$\Phi_i = \theta(\mathbf{x}_{13}, \mathbf{x}_{24}) - \theta_0(\mathbf{X}_{13}, \mathbf{X}_{24}) \quad (4.103)$$

$$\Phi_i = \theta(\mathbf{x}_{13}, \mathbf{x}_{25}) - \theta_0(\mathbf{X}_{13}, \mathbf{X}_{25}) \quad (4.104)$$

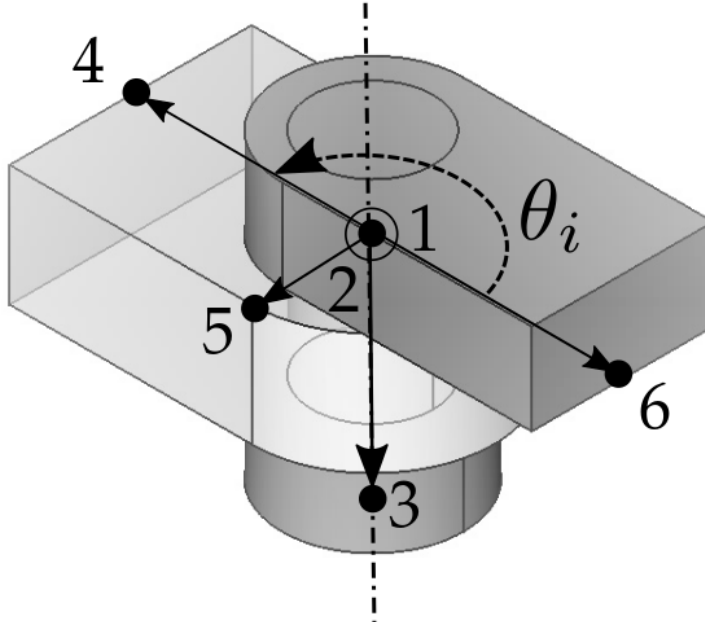


Figure 4.1: 3D Revolute joint

Another point is defined in a plane perpendicular to the revolution axis containing the point 4. The angle of rotation can then be calculated using equation 4.101, with:

$$\theta_i = \theta(\mathbf{x}_{16}, \mathbf{x}_{24}) \quad (4.105)$$

C. 3D cylindrical joint

The joint has two degrees of freedom: the same rotational as before, and a relative slide between the bodies along the rotation axis. Five points are also used for the constraints, three aligned in the axis, and a fourth and fifth away from it:

- The first three are used for an alignment constraint of points in space:

$$\Phi_i = \mathbf{x}_{12} \times \mathbf{x}_{13} \quad (4.106)$$

- The same constant orientation vectors are defined as in the revolute joint:

$$\Phi_i = \theta(\mathbf{x}_{13}, \mathbf{x}_{24}) - \theta_0(\mathbf{X}_{13}, \mathbf{X}_{24}) \quad (4.107)$$

$$\Phi_i = \theta(\mathbf{x}_{13}, \mathbf{x}_{25}) - \theta_0(\mathbf{X}_{13}, \mathbf{X}_{25}) \quad (4.108)$$

The angle of rotation can then be calculated adding a sixth point, and using equation 4.105.

For measuring the sliding in the joint, the relative movement between coaxial points is used:

$$L_i = \mathbf{x}_2 - \mathbf{x}_1 = \sum_{j=1}^{SP2} N_j(\mathbf{X}^2) \begin{pmatrix} q_{1j}(t) \\ q_{2j}(t) \\ q_{3j}(t) \end{pmatrix} - \sum_{k=1}^{SP1} N_k(\mathbf{X}^1) \begin{pmatrix} q_{1k}(t) \\ q_{2k}(t) \\ q_{3k}(t) \end{pmatrix} \quad (4.109)$$

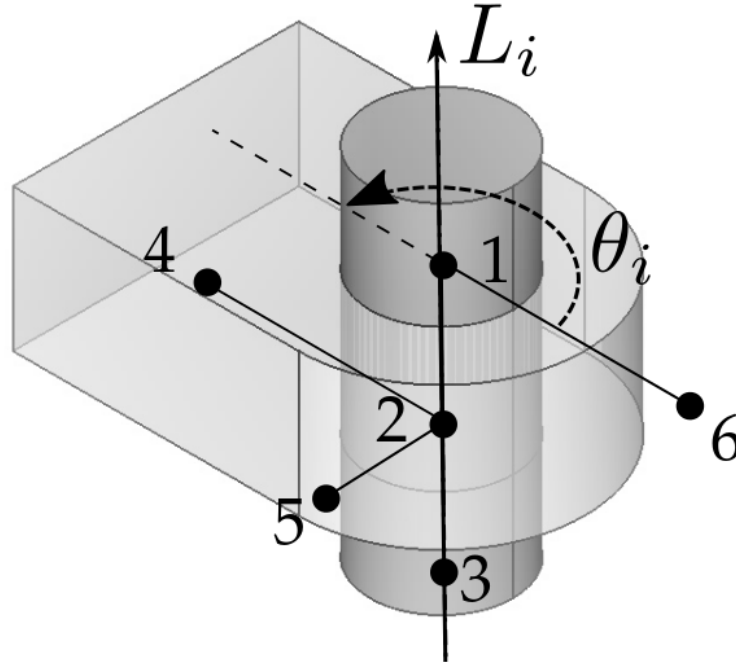


Figure 4.2: 3D Cylindrical joint

D. 3D prismatic joint

Finally, the one degree of freedom slider can be built using the same points as before. The main difference is that the relative angle θ_i is fixed, so the angle measurement is converted to a new constraint:

$$\Phi_i = \theta(\mathbf{x}_{16}, \mathbf{x}_{24}) - \theta_0(\mathbf{X}_{16}, \mathbf{X}_{24}) \quad (4.110)$$

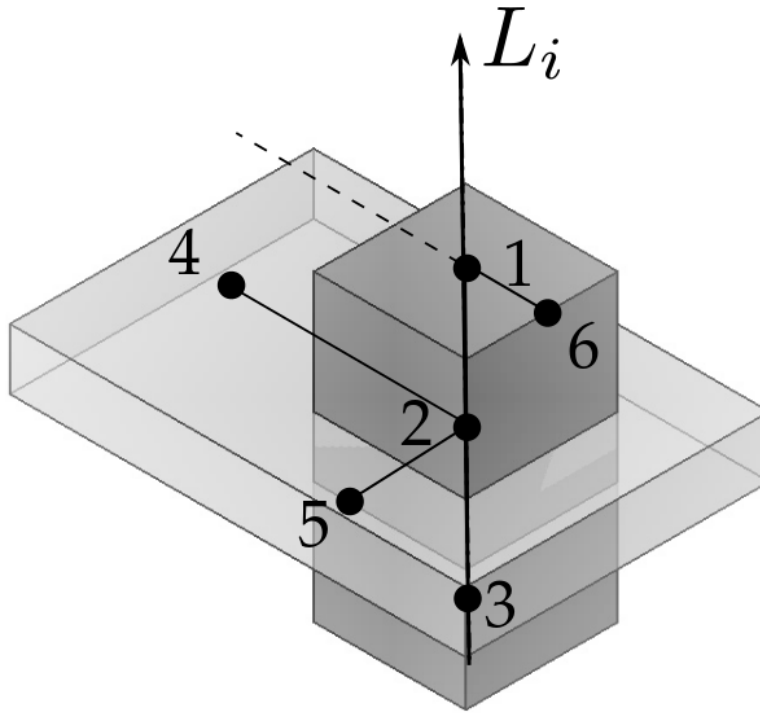


Figure 4.3: 3D Prismatic joint

4.5 Generalized contact formulation

The algorithms involved in the construction of the meshfree shape functions are also used to formulate an automatic contact detection method. Let's recall how the support for the shape functions is computed:

- A set of scattered nodes is defined.
- A boundary is discretized by means of cells, defined by their vertex points.
- Each cell defines the location of its Gauss points (GPs).
- The GPs are iterated and each one searches for its neighbour nodes, adding to its support those that reside within a distance defined by the product of the average nodal distance field surrounding it d_c , and the dimensionless parameter α_i .

The optimization of last step involves creating structures that group the domain nodes depending on their location. The most popular approach for speeding up the node searching computation is the Octree voxel grouping [54]. It consists in a subdivision of the 3D space in blocks of eight equally sized boxes. Each one is recursively subdivided until a predefined maximum number of nodes are located inside of each box. The analogue technique in 2D receives the name of quadtree.

Once the nodes are quickly found, their position can be evaluated. For constructing the shape function values at the GPs, the distance between this point and each node is used. But, if the relative distance between nodes is calculated, then they can be connected using a similar criterion. This is the principle of the Delaunay triangularization constrained by α -shapes [1].

For each time step, the relative distance of the boundary nodes is evaluated. Once a set of three nodes (or four in 3D) are close enough (as defined by the α parameter), the connectivity of a contact triangular element (or tetrahedron) is defined. For simplicity's sake the 2D approach is presented here, although as the method is distance based, its extension to 3D is straightforward.

Let a contact element be formed by the nodes (P_0, P_1, P_2) , where the pair (P_0, P_1) corresponds to nodes within the same boundary.

Let $P_i = \{x_i, y_i\}$ be the 2D components of the nodal positions. The normal unit vector of the contact element can be calculated as:

$$\mathbf{n} = \left\{ \begin{array}{c} -(y_1 - y_0) \\ x_1 - x_0 \end{array} \right\} \frac{1}{\sqrt{(y_1 - y_0)^2 + (x_1 - x_0)^2}}$$

The gap, g , or contact distance is defined as:

$$\begin{aligned} g &= (\mathbf{P}_2 - \mathbf{P}_0) \cdot \mathbf{n} \\ &= (\mathbf{P}_2 - \mathbf{P}_1) \cdot \mathbf{n} \\ &= \frac{(\mathbf{P}_2 - \mathbf{P}_1) + (\mathbf{P}_2 - \mathbf{P}_0)}{2} \cdot \mathbf{n} \end{aligned}$$

Or, in components:

$$g = \frac{0.5 ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))}{\sqrt{(y_1 - y_0)^2 + (x_1 - x_0)^2}}$$

A contact potential is defined, Φ , so that when the gap turns negative, $g < 0$, the contact force is proportional to that gap:

$$\Phi = \frac{0.25 ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))^2}{(y_1 - y_0)^2 + (x_1 - x_0)^2}$$

The previous function defines a temporary constraint between the nodes of the system. Considering variations of the normal vector, the derivative formulation is quite involved, as detailed in Appendix A.

In the case that the normal vector can be set as a constant within the nonlinear solving step, the resulting equations get greatly simplified. The components of the

gap are reduced to:

$$g = n_y \left(y_2 - \frac{y_1 + y_0}{2} \right) + n_x \left(x_2 - \frac{x_1 + x_0}{2} \right)$$

And the generalized constraint force is then:

$$\Phi = g^2 = \left(n_y \left(y_2 - \frac{y_1 + y_0}{2} \right) + n_x \left(x_2 - \frac{x_1 + x_0}{2} \right) \right)^2$$

The derivatives of the contact force components are needed for the calculation of the residual contributions:

$$\frac{\partial \Phi}{\partial x_0} = -n_x \left(n_y \left(y_2 - \frac{y_1 + y_0}{2} \right) + n_x \left(x_2 - \frac{x_1 + x_0}{2} \right) \right)$$

$$\frac{\partial \Phi}{\partial y_0} = -n_y \left(n_y \left(y_2 - \frac{y_1 + y_0}{2} \right) + n_x \left(x_2 - \frac{x_1 + x_0}{2} \right) \right)$$

$$\frac{\partial \Phi}{\partial x_1} = -n_x \left(n_y \left(y_2 - \frac{y_1 + y_0}{2} \right) + n_x \left(x_2 - \frac{x_1 + x_0}{2} \right) \right)$$

$$\frac{\partial \Phi}{\partial y_1} = -n_y \left(n_y \left(y_2 - \frac{y_1 + y_0}{2} \right) + n_x \left(x_2 - \frac{x_1 + x_0}{2} \right) \right)$$

$$\frac{\partial \Phi}{\partial x_2} = 2n_x \left(n_y \left(y_2 - \frac{y_1 + y_0}{2} \right) + n_x \left(x_2 - \frac{x_1 + x_0}{2} \right) \right)$$

$$\frac{\partial \Phi}{\partial y_2} = 2n_y \left(n_y \left(y_2 - \frac{y_1 + y_0}{2} \right) + n_x \left(x_2 - \frac{x_1 + x_0}{2} \right) \right)$$

And finally, the second derivative, required for the nonlinear Newton-Raphson solver, can be written in the following matrix form:

$$\nabla^2 \Phi = \begin{pmatrix} \frac{n_x^2}{2} & \frac{n_x n_y}{2} & \frac{n_x^2}{2} & \frac{n_x n_y}{2} & -n_x^2 & -n_x n_y \\ \frac{n_x n_y}{2} & \frac{n_y^2}{2} & \frac{n_x n_y}{2} & \frac{n_y^2}{2} & -n_x n_y & -n_y^2 \\ \frac{n_x^2}{2} & \frac{n_x n_y}{2} & \frac{n_x^2}{2} & \frac{n_x n_y}{2} & -n_x^2 & -n_x n_y \\ \frac{n_x n_y}{2} & \frac{n_y^2}{2} & \frac{n_x n_y}{2} & \frac{n_y^2}{2} & -n_x n_y & -n_y^2 \\ -n_x^2 & -n_x n_y & -n_x^2 & -n_x n_y & 2n_x^2 & 2n_x n_y \\ -n_x n_y & -n_y^2 & -n_x n_y & -n_y^2 & 2n_x n_y & 2n_y^2 \end{pmatrix}$$

4.6 Proposed multibody formulation

The weak form of the equations of motion for a system composed of both rigid and flexible bodies is obtained adding the weak equations of the rigid and the deformable subsystems and a coupling term that account for the effect of the joints that connect both subsystems, imposing a set of Φ_{RD} constraints. This coupling term is the virtual work of the constraint forces acting over the rigid and deformable components $\mathbf{f}_{\Phi_{RD}}$ and $\mathbf{f}_{\Phi_{DR}}$ respectively. Thermal effects are also taken into account in the weak form for any rigid or flexible continuous bodies. The weakly couple includes thermal deformation, and also configuration dependent heat fluxes. For simplification, the terms for internal and external discretized forces are grouped in \mathbf{f}_R^h and \mathbf{f}_D^h , as well as the ones for internal and external heat fluxes in \mathbf{Q}^h .

Based on these considerations, the global weak form for the thermo-mechanical equations of the system is:

$$\begin{aligned} & \delta \mathbf{q}_R^T \cdot (\mathbf{M}_R \ddot{\mathbf{q}}_R) + \delta \mathbf{q}_D^T \cdot (\mathbf{M}_D \ddot{\mathbf{q}}_D) - \delta \mathbf{q}_R^T \cdot \mathbf{f}_R^h - \delta \mathbf{q}_D^T \cdot \mathbf{f}_D^h \\ & + \delta \mathbf{q}_R^T \cdot [\mathbf{f}_{\Phi_R} + \mathbf{f}_{\Phi_{RD}}] + \delta \mathbf{q}_D^T \cdot [\mathbf{f}_{\Phi_D} + \mathbf{f}_{\Phi_{DR}}] \\ & + \delta \mathbf{q}_T^T (\mathbf{C} \dot{\mathbf{q}}_T) - \delta \mathbf{q}_T^T (\mathbf{Q}^h) + \delta \mathbf{q}_T^T (\mathbf{Q}_{\Phi}) = 0 \quad \forall \delta \mathbf{q}_R, \delta \mathbf{q}_D, \delta \mathbf{q}_T \end{aligned} \quad (4.111)$$

assuming that:

\mathbf{q}_R and \mathbf{q}_D satisfy the mechanical constraints $\Phi_R, \Phi_D, \Phi_{RD}$, and

\mathbf{q}_T satisfies the thermal constraints Φ_T .

Collecting all motion coordinates under vector $\mathbf{q}_M = \{\mathbf{q}_R, \mathbf{q}_D\}^T$, the equations can be written in a compact matrix format as:

$$\underbrace{\begin{pmatrix} \mathbf{M}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_D \end{pmatrix}}_{\mathbf{M}} \cdot \underbrace{\begin{pmatrix} \ddot{\mathbf{q}}_R \\ \ddot{\mathbf{q}}_D \end{pmatrix}}_{\ddot{\mathbf{q}}_M} - \underbrace{\begin{pmatrix} \mathbf{f}_R^h \\ \mathbf{f}_D^h \end{pmatrix}}_{\mathbf{f}^h} + \underbrace{\begin{pmatrix} \mathbf{f}_{\Phi_R} \\ \mathbf{f}_{\Phi_D} \end{pmatrix} + \begin{pmatrix} \mathbf{f}_{\Phi_{RD}} \\ \mathbf{f}_{\Phi_{DR}} \end{pmatrix}}_{\mathbf{f}_{\Phi_M}} = \mathbf{0} \quad (4.112)$$

$$\mathbf{C} \cdot \dot{\mathbf{q}}_T - \mathbf{Q}^h + \mathbf{Q}_{\Phi_T} = \mathbf{0}$$

Φ denoting the global set of constraints over the entire flexible system. If the Lagrange multiplier method is employed, the constraint force vector takes the form $\mathbf{f}_{\Phi_M} = \mathbf{G}^T \lambda_M$, the constraint flux $\mathbf{Q}_{\Phi_T} = \mathbf{T}^T \lambda_T$, and the constraint set Φ must be appended to the differential equation system (4.112) to define an index-3 DAE that must be solved for $(\mathbf{q}(t), \lambda(t))$. If a penalty method is employed, the constraint force and flux take the form $\mathbf{f}_{\Phi} = \mathbf{G}^T \alpha_M \Phi_M$, $\mathbf{Q}_{\Phi} = \mathbf{T}^T \alpha_T \Phi_T$ respectively, and the differential system completely defines the equilibrium:

$$\underbrace{\begin{pmatrix} \mathbf{M}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_D \end{pmatrix}}_{\mathbf{M}} \cdot \underbrace{\begin{pmatrix} \ddot{\mathbf{q}}_R \\ \ddot{\mathbf{q}}_D \end{pmatrix}}_{\ddot{\mathbf{q}}_M} + \underbrace{\begin{pmatrix} \mathbf{0} \\ \mathbf{f}_{int_D}^h \end{pmatrix}}_{\mathbf{f}_{int}^h} - \mathbf{f}_{ext}^h + \mathbf{G}^T \boldsymbol{\alpha}_M \boldsymbol{\Phi}_M = \mathbf{0} \quad (4.113)$$

$$\mathbf{C} \cdot \dot{\mathbf{q}}_T + \mathbf{H} \mathbf{q}_T - \mathbf{Q}_{ext}^h + \mathbf{T}^T \boldsymbol{\alpha}_T \boldsymbol{\Phi}_T = \mathbf{0}$$

where:

\mathbf{M} is the global mass matrix, logically partitioned into rigid and flexible contributions from the system:

\mathbf{M}_R Rigid bodies' mass matrix, composed by individual masses and continuous bodies local matrices:

$$\mathbf{M}_R = \mathbf{A}_{b=1}^{n_{bodies}} [M_{IJ}]_R = \mathbf{A}_{b=1}^{n_{bodies}} \left[m_I \delta_{IJ} + \int_{B_0} \rho_b C_J C_I dV \right]$$

\mathbf{M}_D Flexible bodies' mass matrix, assembled from local matrices described at each Gauss point with overlapping domain of influence:

$$\mathbf{M}_D = \mathbf{A}_{p=1}^{n_{GPs}} [M_{IJ}]_D = \mathbf{A}_{p=1}^{n_{GPs}} \left[\int_{B_0} \rho_p N_J N_I \mathbb{I}_{dim} dV \right]$$

\mathbf{f}_{int}^h is the discretized global internal forces vector of the flexible bodies:

$$\mathbf{f}_{int}^h = \mathbf{A}_{p=1}^{n_{GPs}} \{ \mathbf{f}_p^h \}_I = \mathbf{A}_{p=1}^{n_{GPs}} \left\{ \int_{B_0} \mathbf{P}_p \cdot \nabla_0 N_I dV \right\}$$

\mathbf{f}_{ext}^h is the discretized global external forces vector, integrated in the domain and boundary:

$$\mathbf{f}_{ext}^h = \mathbf{A}_{p=1}^{n_{GPs}} \{ \mathbf{f}_p^h \}_I + \mathbf{A}_{q=1}^{n_{GPs}} \{ \mathbf{f}_q^h \}_J = \mathbf{A}_{p=1}^{n_{GPs}} \left\{ \int_{B_0} N_I \cdot \mathbf{b}_0 dV \right\} + \mathbf{A}_{q=1}^{n_{GPs}} \left\{ \int_{\partial B_0} N_J \cdot \mathbf{t}_0 dS \right\}$$

\mathbf{C} is the global capacitive matrix of the continuous thermal bodies:

$$\mathbf{C} = \mathbf{A}_{p=1}^{n_{GPs}} [C_{IJ}] = \mathbf{A}_{p=1}^{n_{GPs}} \left\{ \int_{B_0} \rho C_p N_J N_I dV \right\}$$

\mathbf{H} is the discretized global thermal conductivity matrix of the system:

$$\mathbf{H} = \mathbf{A}_{p=1}^{n_{GPs}} [H_{IJ}] = \mathbf{A}_{p=1}^{n_{GPs}} \left\{ \int_{B_0} \nabla_0^T N_J F_p^{-1} \kappa_0 F_p^{-T} \nabla_0 N_I dV \right\}$$

\mathbf{Q}_{ext}^h is the discretized global external heat loads vector, composed of internal generation and heat fluxes:

$$\mathbf{Q}_{ext}^h = \mathbf{A}_{p=1}^{n_{GPs}} \{ \mathbf{Q}_p^h \}_I + \mathbf{A}_{q=1}^{n_{GPs}} \{ \mathbf{Q}_q^h \}_J = \mathbf{A}_{p=1}^{n_{GPs}} \left\{ \int_{B_0} N_I \cdot r_0 dV \right\} + \mathbf{A}_{q=1}^{n_{GPs}} \left\{ \int_{\partial B_0} N_J \cdot \mathbf{q}_0 \cdot \mathbf{n} dS \right\}$$

$\boldsymbol{\Phi}_M, \boldsymbol{\Phi}_T$ are the motion and thermal constraints, respectively.

\mathbf{G}, \mathbf{T} are the gradient of the constraints:

$$\mathbf{G} = \nabla_{\mathbf{q}_M} \boldsymbol{\Phi}_M$$

$$\mathbf{T} = \nabla_{\mathbf{q}_T} \boldsymbol{\Phi}_T$$

$\boldsymbol{\alpha}_M, \boldsymbol{\alpha}_T$ are the penalty vectors for the constraints.

This section describes simulations of increasing complexity computed with the developed multibody code and how they correlate to the analytical solution. FEM and meshfree methods have been implemented in the code, and both have been verified using the Ansys FEM code, and also by step-by-step scripting of the internal matrices for the simple cases (detailed in appendix B).

5.1 3-node triangle

The most simple case is a 2D plane strain triangle defined only by a set of nodes in each of the vertices, as shown in figure 5.1. The sides are given a value $h = 1$, and the displacement of both left nodes is constrained. Node 1 is located at the origin of the coordinate system.

The elastic material properties have been set to the following values, both for linear Hooke, and nonlinear Saint-Venant Kirchhoff models:

- Density: $\rho = 7850$
- Young's modulus: $E = 0.7E6$
- Poisson coefficient: $\nu = 0.3$

5.1.1 Shape functions

The area can be integrated using only one Gauss Point (GP) for all the three methods evaluated. This GP is located in $(1/3, 1/3)$. Nevertheless, in general the shape functions of the meshfree methods are of higher order and may take advantage of an increased number of Gauss Points. For calculating the Mass and Thermal

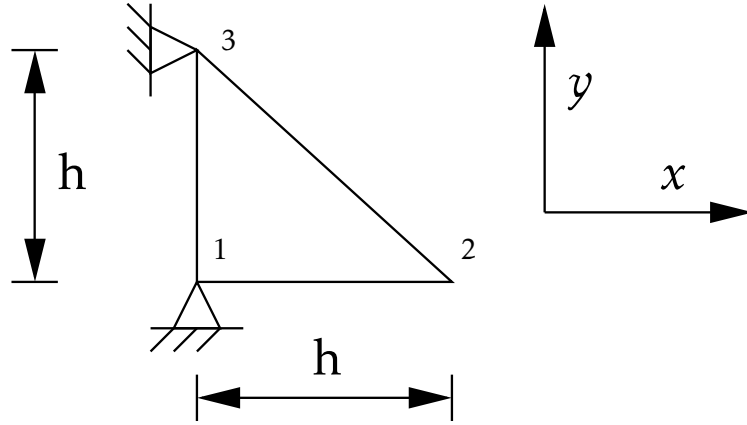


Figure 5.1: three node triangle

Capacity matrices, three Gauss Points are used, with locations $(1/6, 1/6)$, $(2/3, 1/6)$, $(1/6, 2/3)$, and equal weights of $1/3$.

All of the shape function information is given in a matrix form. Each matrix gives the values for the shape function in the first row, meanwhile the partial derivatives values are ordered in the first $(\partial/\partial x)$ and second $(\partial/\partial y)$ rows:

$$\mathcal{N}_i = \begin{bmatrix} N_i^T \\ \partial N_i^T / \partial x \\ \partial N_i^T / \partial y \end{bmatrix}$$

A. FEM

The type of element used is the regular CST (constant strain) triangle. The shape functions are linear polynomials, deriving to a constant gradient for the whole element. Of course, the stiffness matrix is integrated exactly by just one Gauss Point. The shape function and derivatives are therefore:

$$\mathcal{N}_1(1/3, 1/3) = \begin{bmatrix} 0.333333 & 0.333333 & 0.333333 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

The three shape functions associated to each of the nodes used to compute the Mass and Thermal Capacity matrices are represented by the following arrays:

$$\mathcal{N}_1(1/6, 1/6) = \begin{bmatrix} 0.666667 & 0.166667 & 0.166667 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\mathcal{N}_2(2/3, 1/6) = \begin{bmatrix} 0.166667 & 0.666667 & 0.166667 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\mathcal{N}_3(1/6, 2/3) = \begin{bmatrix} 0.166667 & 0.166667 & 0.666667 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

B. Radial Basis Functions interpolation

A basic radial multiquadrics (MQ) basis functions evaluation is presented. No polynomial augmented basis is applied, and the functions parameters correspond to the following:

- $\beta = q = 1/2$, for the original RBFs proposed by [19].
- $d_c = 1.13807$ is the average distance between the nodes
- $\alpha_i = 3.5$, leads to an equivalent value of $c = \alpha_i d_c = 3.983245$.

Using one Gauss Point, the shape function does not lead any more to a constant coefficient, as was the case for FEM:

$$\mathcal{N}_1(1/3, 1/3) = \begin{bmatrix} 0.343836 & 0.321592 & 0.321592 \\ -0.990709 & 0.997061 & -0.0159348 \\ -0.990709 & -0.0159348 & 0.997061 \end{bmatrix}$$

It's worth that as expected, the truncation of the RBF in a finite local domain produces an integration error of the domain shape. Adding all the values for the shape function we arrive to a value of:

$$\sum_{i=1}^3 N_i = 0.987019 < 1$$

The truncation error in this case is heavily dependent on the parameter α_i , due to a very limited number of support nodes. For example, a general rule to define a parameter-less RBF is based in the following procedure:

1. The neighbouring nodes are found using the maximum distance $c = \alpha_i d_c$,
2. The average distance d_{ij} between the GP and the NP neighbours is used to redefine the local size relative factor:

$$\alpha_i = \frac{\sum_{j=1}^{NP} d_{ij}}{d_c NP}$$

When applied to a low number of support nodes, as in this case, the resulting shape function is heavily truncated. The corresponding average distances define $\alpha_i = 0.654929$. The moment matrix with only three points has a high condition number, which in practice produces increases the interpolation error. The corresponding shape function is:

$$\mathcal{N}_1(1/3, 1/3) = \begin{bmatrix} 0.412152 & 0.230399 & 0.230399 \\ -0.938652 & 0.987919 & -0.123922 \\ -0.938652 & -0.123922 & 0.987919 \end{bmatrix}$$

And the error in the shape function is no longer small:

$$\sum_{i=1}^3 N_i = 0.872951 \ll 1$$

Returning to the previous $\alpha_i = 3.5$, in the case of considering three Gauss Points for the cell integration, the values of the respective shape functions are the following:

$$\mathcal{N}_1(1/6, 1/6) = \begin{bmatrix} 0.673551 & 0.15916 & 0.15916 \\ -0.985899 & 0.987145 & -0.0207492 \\ -0.985899 & -0.0207492 & 0.987145 \end{bmatrix}$$

$$\mathcal{N}_2(2/3, 1/6) = \begin{bmatrix} 0.174506 & 0.660126 & 0.15485 \\ -1.0026 & 1.00913 & 0.00345126 \\ -0.968617 & -0.0255352 & 0.975039 \end{bmatrix}$$

$$\mathcal{N}_3(1/6, 2/3) = \begin{bmatrix} 0.174506 & 0.15485 & 0.660126 \\ -0.968617 & 0.975039 & -0.0255352 \\ -1.0026 & 0.00345126 & 1.00913 \end{bmatrix}$$

Note that no longer the shape functions are permutations of the same coefficients, as the node 1 is closer to the GPs than the other two. Of course, the symmetry of the triangle translates into symmetric functions for GPs 2,3. Finally, the errors indicate the relation between the quality of the interpolation and the dispersion of the nodes:

$$\sum_{i=1}^3 N_1 = 0.991871$$

$$\sum_{i=1}^3 N_2 = 0.989482$$

$$\sum_{i=1}^3 N_3 = 0.989482$$

C. Moving Least Squares approximation

The MLS approximation studied is obtained using a cubic spline weight function. It degenerates to a linear polynomial shape function when the support is minimal. The parameters used are automatically computed in this case using the previous procedure to rescale α_i :

- $d_c = 1.13807$ is the average distance between the nodes, and
- $\alpha_i = 0.6549295$ is the average distance factor from the GP to all the support nodes.

As previously stated, the results obtained for the three-node MLS triangle are exactly the same as the ones presented for the FEM functions:

$$\mathcal{N}_1(1/3, 1/3) = \begin{bmatrix} 0.333333 & 0.333333 & 0.333333 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

In case that the α_i is not averaged to the GP to nodes distance, a round-off error appears. For example, if we force $\alpha_i = 3.5$ as in the RBF case, we get:

$$\mathcal{N}_1(1/3, 1/3) = \begin{bmatrix} 0.333333 & 0.333333 & 0.333333 \\ -1 & 1 & -1.11022e-16 \\ -1 & 4.16334e-17 & 1 \end{bmatrix}$$

In the case of using three Gauss points letting the code compute the average $\alpha_i = 0.6549295$ again, the same FEM shape functions are produced:

$$\mathcal{N}_1(1/6, 1/6) = \begin{bmatrix} 0.666667 & 0.166667 & 0.166667 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\mathcal{N}_2(2/3, 1/6) = \begin{bmatrix} 0.166667 & 0.666667 & 0.166667 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\mathcal{N}_3(1/6, 2/3) = \begin{bmatrix} 0.166667 & 0.166667 & 0.666667 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

5.1.2 Mechanical validation

A. Gravity acceleration load case

A volumetric gravitational load of magnitude -10 has been applied in the y -axis direction. It is equivalent to a positive vertical $\ddot{y} = 10$ acceleration. Both small and large strain (linear and nonlinear, respectively) formulations have been verified for the simple triangle model, while the procedure, including intermediate and final results have been verified in detail (see Appendix B.1.1).

As expected, same results are obtained for the linear and nonlinear formulations in the case of FEM and EFG shape functions. For the RBF interpolants, the integrated mass is slightly lower, so the deflection of the tip and the internal stress are lower. When the number of integration points is raised to 3 GPs, the behaviour converges for the linear formulation analysis, although the stress distribution is no longer constant, capturing slightly better the variations due to variable section of the geometry, from the tip to the clamped left side. In fact, where the σ_{xy} is the only stress component in the FEM element, other tension effects are captured by the RBF functions. In the case of the nonlinear formulation, results differ more compared to the FEM case, probably due to this effect. Instead of a deformation exclusively related to shear strain, tractions appear along the triangle increasing the model's stiffness.

Table 5.1: Linear results

Method	u_{y1}	σ_{xy}	σ_{VM}
FEM (ANSYS)	-0.09719	-2.617E4	4.53E4
FEM	-0.09719	-2.617E4	4.53E4
RBF 1GP	-0.09306	-2.498E4	4.41E4
RBF 3GPs	-0.09544	-2.539E4	4.45E4
MLS 1GP	-0.09719	-2.617E4	4.53E4

Table 5.2: Nonlinear results

Method	u_{y1}	σ_{VM}
FEM (ANSYS)	-0.09719	4.56E4
FEM	-0.09719	4.56E4
RBF 1GP	-0.06174	4.35E4
RBF 3GPs	-0.06913	4.72E4
MLS	-0.09719	4.56E4

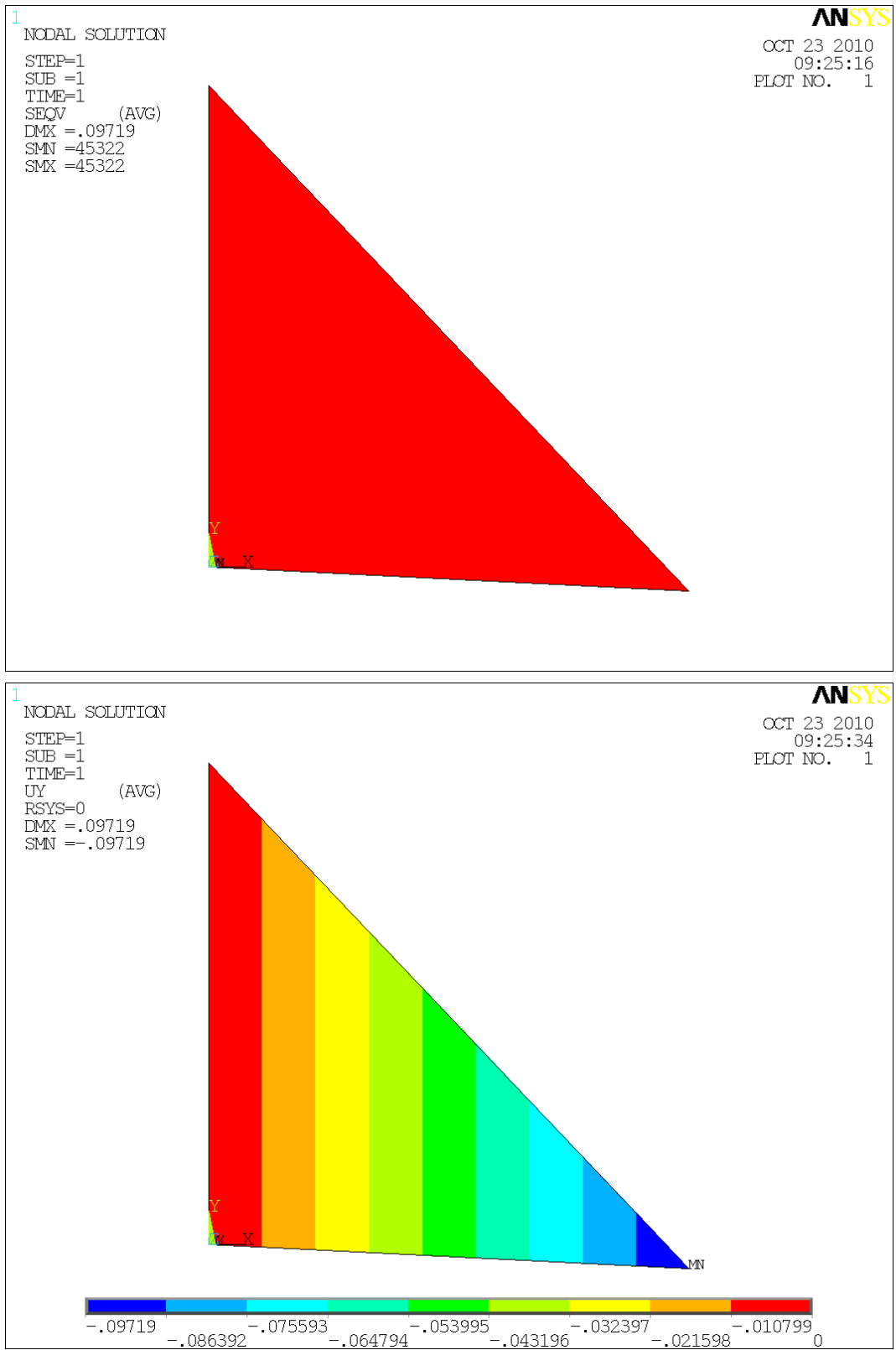


Figure 5.2: Gravitational load linear results in Ansys: Von-Mises stress (top), and vertical displacements (bottom)

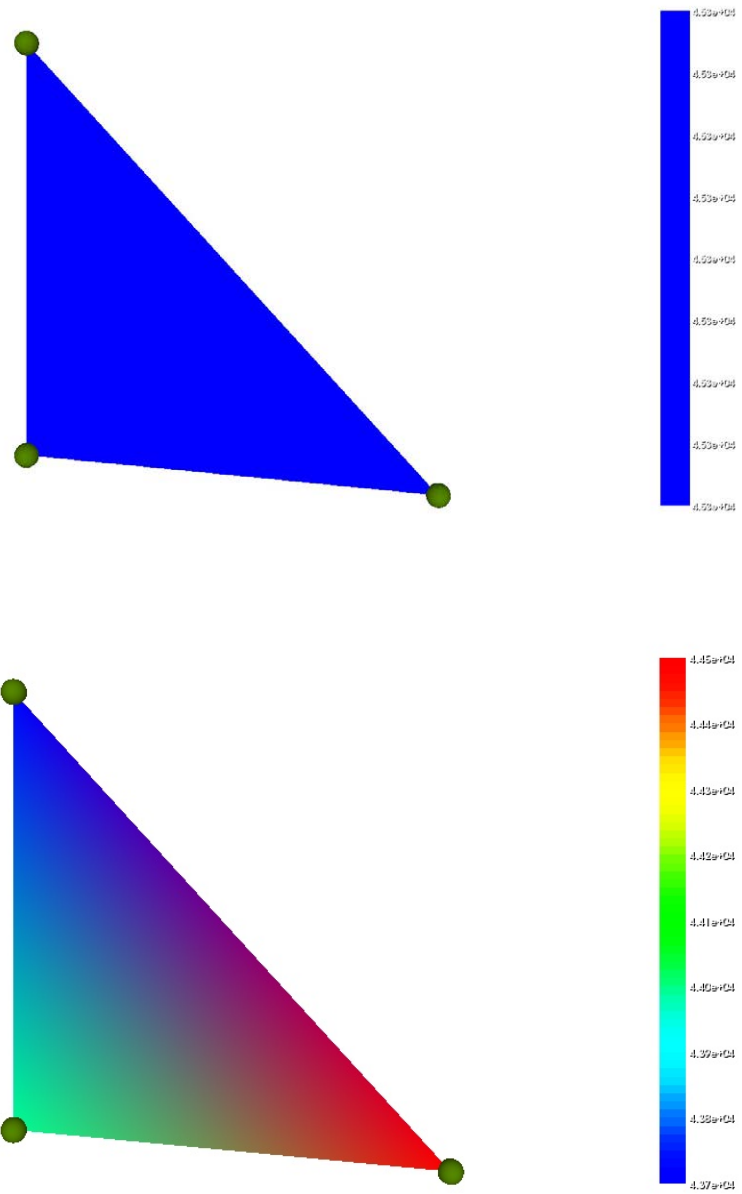


Figure 5.3: Gravitational load linear validation results: Von-Mises stress for FEM (top), and RBFs with 3GPs (bottom)

5.2 4-node square

A second, also simple case of interest consists on a 2D square defined by its vertex nodes (Figure 5.4). The sides are given a value $h = 1$. The support conditions are provided by statically determined attachments: fixed for node 1, and sliding along y -axis for node 3. Both supports are considered thermally isolated.

The thermal and elastic material properties have been set to the following values, both for linear and nonlinear Saint-Venant Kirchhoff thermo-elastic models:

- Thermal conductivity: $\kappa = 224$
- Specific heat capacity: $c_p = 1348$
- Density: $\rho = 7850$
- Young's modulus: $E = 0.7E6$
- Poisson coefficient: $\nu = 0.3$

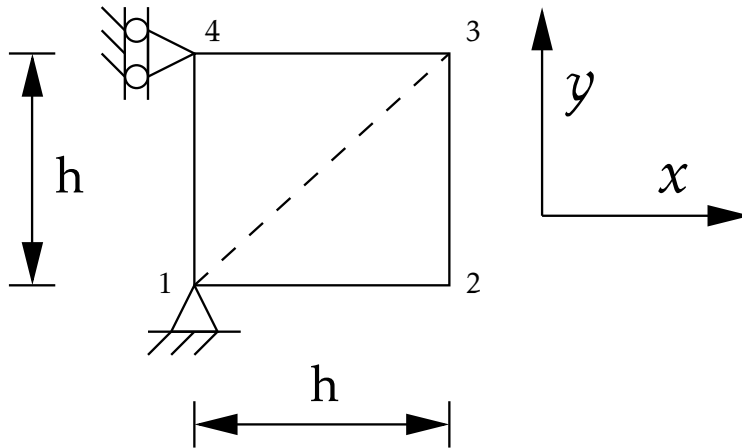


Figure 5.4: three node triangle

This example can be approached and solved using meshfree methods to join to separate triangular domains. Although in this case does not provide any numerical advantage, it allows to validate the general blending algorithm that will be used in following sections. Differences are shown in figure 5.5, where both the square domain and the two joined triangles are shown.

5.2.1 Discretization

A. FEM

Two triangular elements are defined dividing the square by its diagonal between nodes 1 & 4. The shape functions are not numerically identical to previous case, but



Figure 5.5: Different models used: FEM usual single domain (left), and meshfree two domains (right)

for simplicity's sake, their details are avoided due to its similarity. Of course, a 4-node element could be considered for its study, but its not the aim of this research to apply discretization schemes other than triangular grids.

B. Radial Basis Functions

The same division as with FEM is considering but only for the integration of the domain (cells). Each cell is used to defined a set of Gauss Points in the way explained in the previous validation case, i.e. choosing the integration order depending on the accuracy required on the shape function mapping.

Using one GP per cell, the shape functions are of course completely different as with the triangular domain. The parameters defining RBF the interpolation for both points in each centroid of the triangles are the same as before for the case of $\alpha_i = 3.5$. The proximity of the node to the integration point is clear in the shape function values, which are symmetrical as expected:

$$\mathcal{N}_1(2/3, 1/3) = \begin{bmatrix} 0.444458 & 0.217847 & 0.217847 & 0.106776 \\ -0.679781 & 0.676484 & -0.336512 & 0.329943 \\ -0.679781 & -0.336512 & 0.676484 & 0.329943 \end{bmatrix}$$

$$\mathcal{N}_1(1/3, 2/3) = \begin{bmatrix} 0.106776 & 0.217847 & 0.217847 & 0.444458 \\ -0.329943 & 0.336512 & -0.676484 & 0.679781 \\ -0.329943 & -0.676484 & 0.336512 & 0.679781 \end{bmatrix}$$

The addition of one more node to the support domain improves the shape function accuracy. Although adding all the values for each of the shape functions, we arrive to a similar value than in the previous case. This suggests that the truncation of the shape function is more related to the node dispersion:

$$\sum_{i=1}^4 N_i = 0.986928 \simeq 0.987019$$

When three GPs are used for the integration of each cell, the total sum of the RBFs functions correspond to:

$$\sum_{i=1}^4 N_i = 0.991849 \simeq 0.991871$$

$$\sum_{i=1}^4 N_i = 0.989392 \simeq 0.989482$$

C. Moving Least Squares approximation

In the case that the MLS are computed setting α_i with the cell averaging procedure, the additional node will remain just in the limit of the support size and it's shape function contribution will be negligible. This happens with a value of $\alpha_i = 0.937716$, and it can be seen how the shape functions will also degenerate to the FEM shape functions even though the support is no longer minimal:

$$\mathcal{N}_1(2/3, 1/3) = \begin{bmatrix} 0.333333 & 0.333333 & 0.333333 & 0 \\ -1 & 1 & 2.77556e-16 & 0 \\ 4.44089e-16 & -1 & 1 & 0 \end{bmatrix}$$

$$\mathcal{N}_1(1/3, 2/3) = \begin{bmatrix} 0.333333 & 0 & 0.333333 & 0.333333 \\ 8.88178e-16 & 0 & 1 & -1 \\ -1 & 0 & 1.77636e-15 & 1 \end{bmatrix}$$

Obviously for a 1GP integration, using the previous shape functions will lead to exactly the same results as with FEM. In these cases with reduced number of nodes and few integration points—as it happens with the RBFs—it can be preferable to set a higher value for the distance factor. In the case of 3GPs per cell, using the exact α_i parameter is still preferred. If it's force to higher values, the approximation does not improve and weight matrices are generally worse conditioned.

The following richer shape functions are obtained when setting a value of $\alpha_i = 3.5$:

$$\mathcal{N}_1(2/3, 1/3) = \begin{bmatrix} 0.24155 & 0.425116 & 0.24155 & 0.091783 \\ -0.550667 & 0.550667 & 0.449333 & -0.449333 \\ -0.449333 & -0.550667 & 0.550667 & 0.449333 \end{bmatrix}$$

$$\mathcal{N}_1(1/3, 2/3) = \begin{bmatrix} 0.24155 & 0.091783 & 0.24155 & 0.425116 \\ -0.449333 & 0.449333 & 0.550667 & -0.550667 \\ -0.550667 & -0.449333 & 0.449333 & 0.550667 \end{bmatrix}$$

5.2.2 Thermal validation

A dynamic analysis has been performed for a constant absorbed heat flux applied in node 3, with value 5E6. A BDF-1 integration scheme has been used with the following parameters:

- Initial time: $t_0 = 0$
- Step size: $\Delta t = 1\text{E-}3$
- Final time: $t_f = 1\text{E-}2$

The problem has been analysed with all the discretization methods, and compared to Ansys. A complete verification of the different matrices is presented in a dedicated script in the appendix B.2.2.

The short time steps are not a coincidence, but have been chosen specifically to validate adiabatically dominated thermal effects. This, along with the definition of a point load will allow to evaluate the suitability of each shape function, when the type of loads are prone to produce singularities.

Results summarized in table 5.3 show that the C^∞ are rich enough with only four nodes to capture this localized effect, at least slightly better.

Table 5.3: Thermal transient results

Method	$T_2(t = 1\text{E-}2)$
FEM (ANSYS)	0.063586
FEM	0.063586
RBF 1GP	0.085523
RBF 3GPs	0.097295
MLS 1GP	0.063586
MLS 1GP ($\alpha_i = 3.5$)	0.087747
MLS 3GPs ($\alpha_i = 3.5$)	0.095379

5.2.3 Mechanical validation

A. Constant strain: Patch test

Although not neither necessary nor sufficient for convergence [5], the patch test is an well known and used test as an initial test of newly proposed finite elements. It's clear from the truncation of the shape function, that the RBF based discrete

model will not pass the test. As it will be shown in the mechanical benchmark section 6.1, this does not directly imply any lack of accuracy when simulating flexible systems.

The problem is stated as follows: The side between nodes 2 and 3 is subjected to a constant traction with a value of $2E4$ per unit of length. Considering there are only two nodes defining the boundary segment, this is equivalent to the application of a force of $1E4$ in each node.

The results in table 5.4 show that, as expected, the meshfree methods struggle mapping exactly the constant strain field, although the error is low. In the case of the MLS, low support size helps in getting the exact result but same effect would appear with denser meshes.

Table 5.4: Linear results

Method	ux_{2-3}	σ_x
FEM (ANSYS)	0.028572	2.00E4
FEM	0.028572	2.00E4
RBF 1GP	0.027874	1.97E4
RBF 3GPs	0.028243	1.99E4
MLS	0.028572	2.00E4

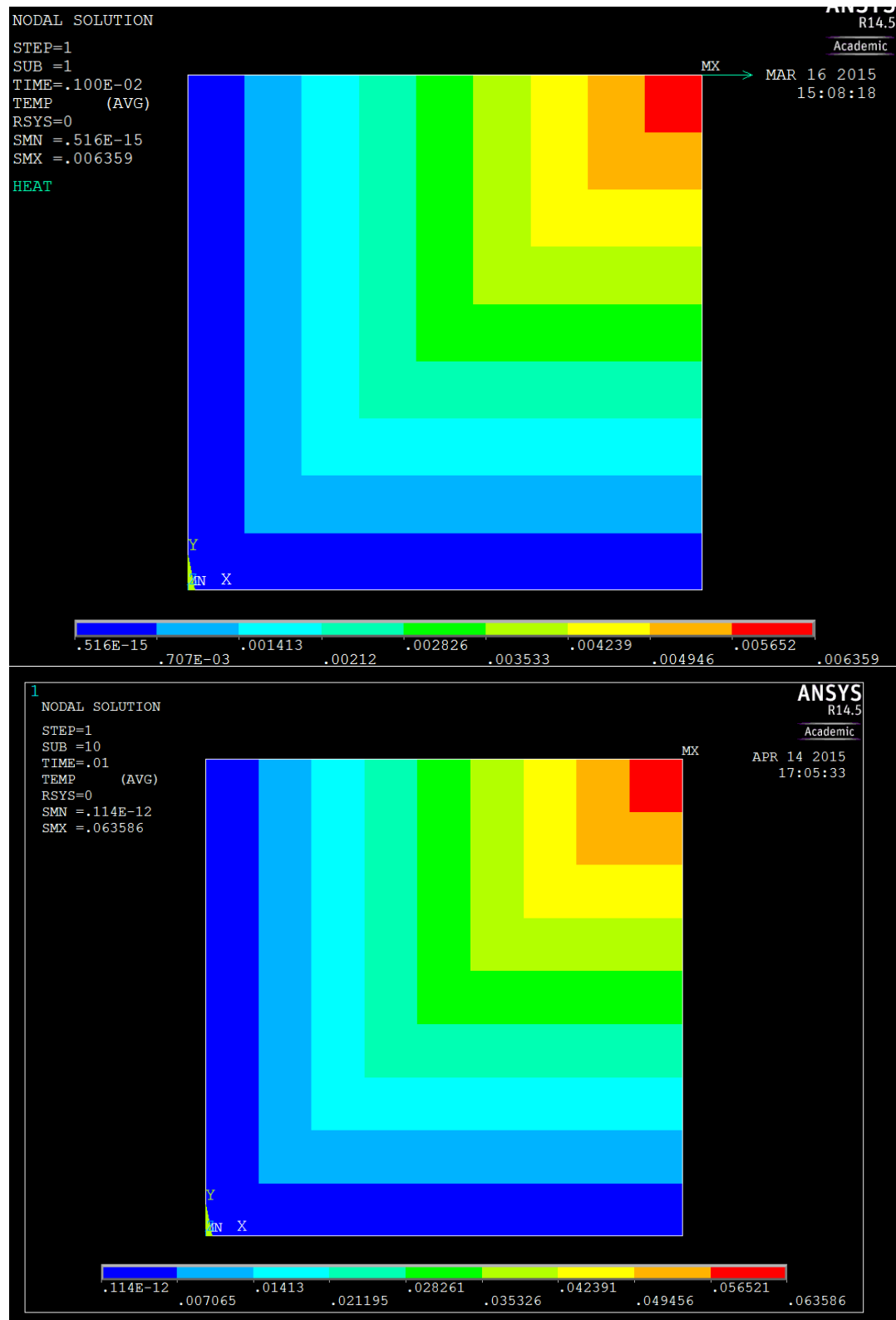


Figure 5.6: Temperature evolution for Ansys FEM model: first step at $t = 1\text{E-}3$ (top), and last step at $t = 1\text{E-}2$ (bottom)

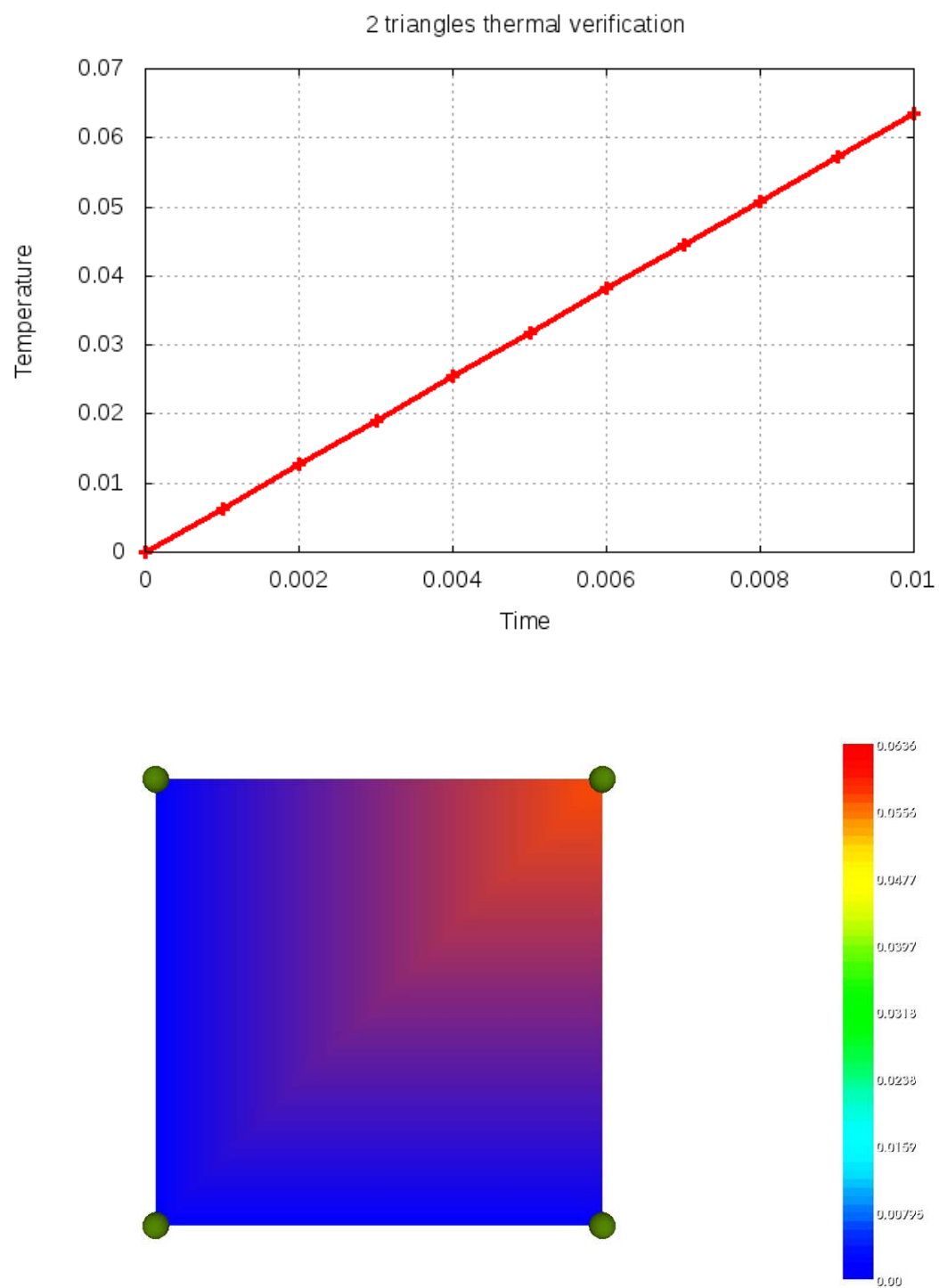


Figure 5.7: Temperature evolution for simulated FEM model: Node 3 (top), and last step at $t = 1\text{E-}2$ (bottom)

5.3 Prismatic beam

A 3D cantilever beam (Figure 5.9) is considered in this section, schematically shown in Figure 5.8. It has a square hollow cross-section, and a relatively thick wall thickness. The parameters that define its geometry are the following:

- Length of beam, $L = 40$,
- Height and width, $h_e = 8$, and
- Internal hollow height and width, $h_i = 6$, corresponding to a thickness of $e = 1$.

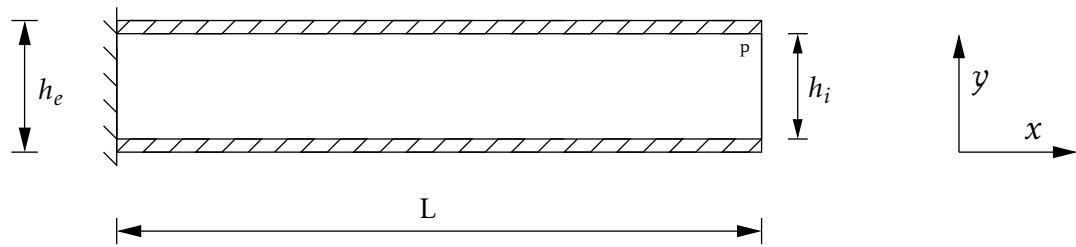


Figure 5.8: Prismatic beam section view

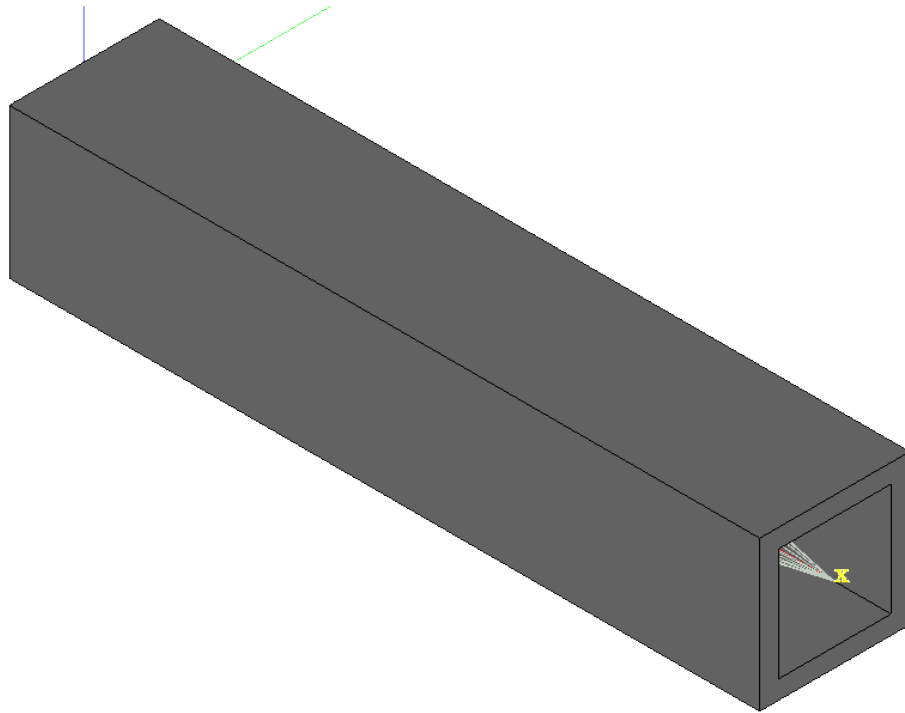


Figure 5.9: 3D beam geometry

The objective is analysing the effects on a 3D beam when statically loaded with force of magnitude $F_P = 8$ applied in the centre of its top flange (point P in Figure 5.8).

Once again, the same elastic material properties have been set for both linear and nonlinear SVK elastic constitutive models:

- Density: $\rho = 7850$
- Young's modulus: $E = 0.7E6$
- Poisson coefficient: $\nu = 0.3$

5.3.1 Beam discrete models

A. 3D solid beams

Two different models have been generated for the 3D real beam:

- Reference model with $> 300k$ nodes (Figure 5.10). It will be solved in ANSYS, in order to evaluate the error of the other coarse discretizations.
- Coarse model with 261 nodes (Figure 5.11). Used for comparing the accuracy between FEM and meshfree methods in 3D tetrahedral grids with flatten elements/cells (Figure 5.12).

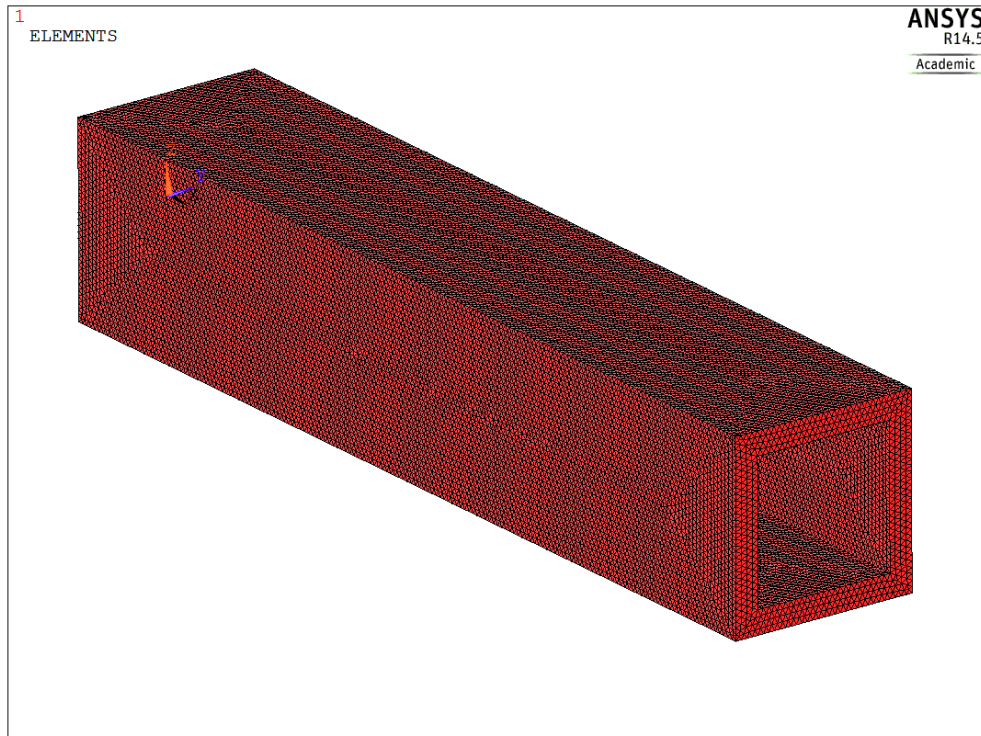


Figure 5.10: 3D reference mesh model in ANSYS

The reference results show a non-negligible local effect of the point load (Figure 5.13). The maximum deformation in the point of application of the load is $U_{y-max} =$

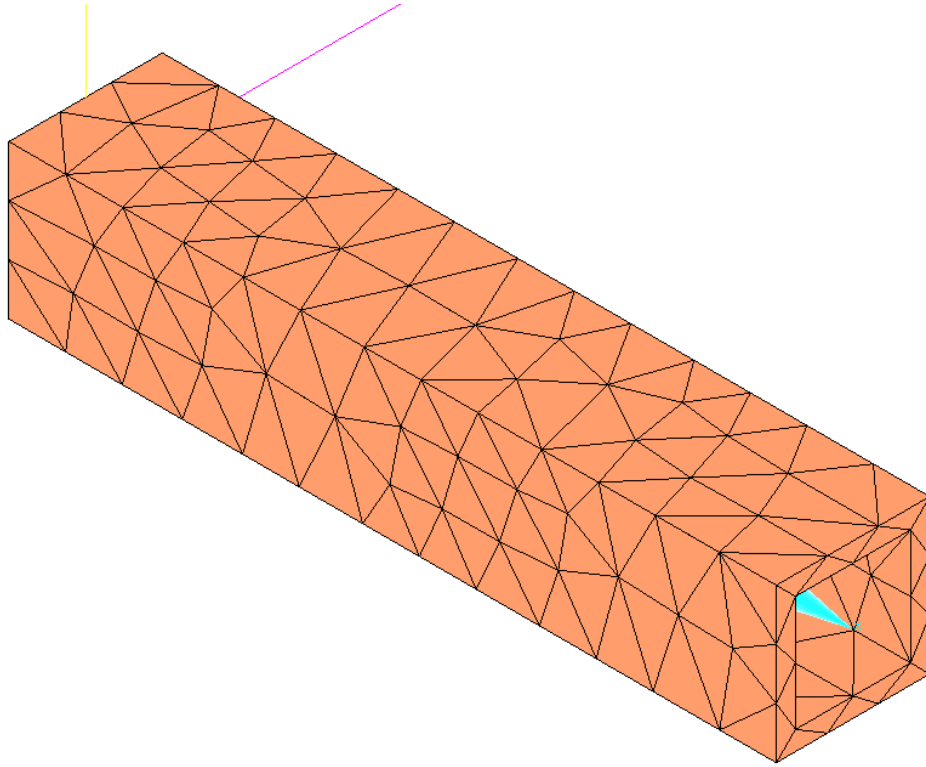


Figure 5.11: 3D mesh model

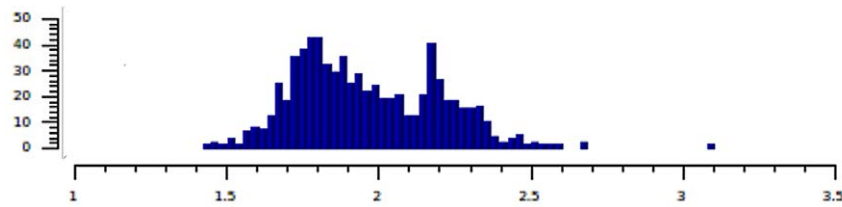


Figure 5.12: Aspect ratio histogram for 3D coarse beam model elements

-1.496 , while the deformation in the upper central point is $U_{y-top} = -1.39$. The maximum deformation is due to the numerical singularity being capture by the very fine mesh. As we are interested in validating the global behaviour of the beam, the later value, far enough from the singularity but still affected by de upper flange deformation will be used as the reference solution for the upper flange.

The lower flange mid-point has a vertical deformation of $U_{y-bot} = -1.05$. It also will be used for validation to other models as it depends only on the global bulk behaviour of the beam.

The 3D coarse model has been solved both with FEM and meshfree RBF functions. As expected, both show higher stiffness due to the low number of nodes and elements used to approximate the deformation. This stiffer behaviour is even worse in the FEM case, as it's also influenced by the high aspect ratio (Figure 5.12). It's

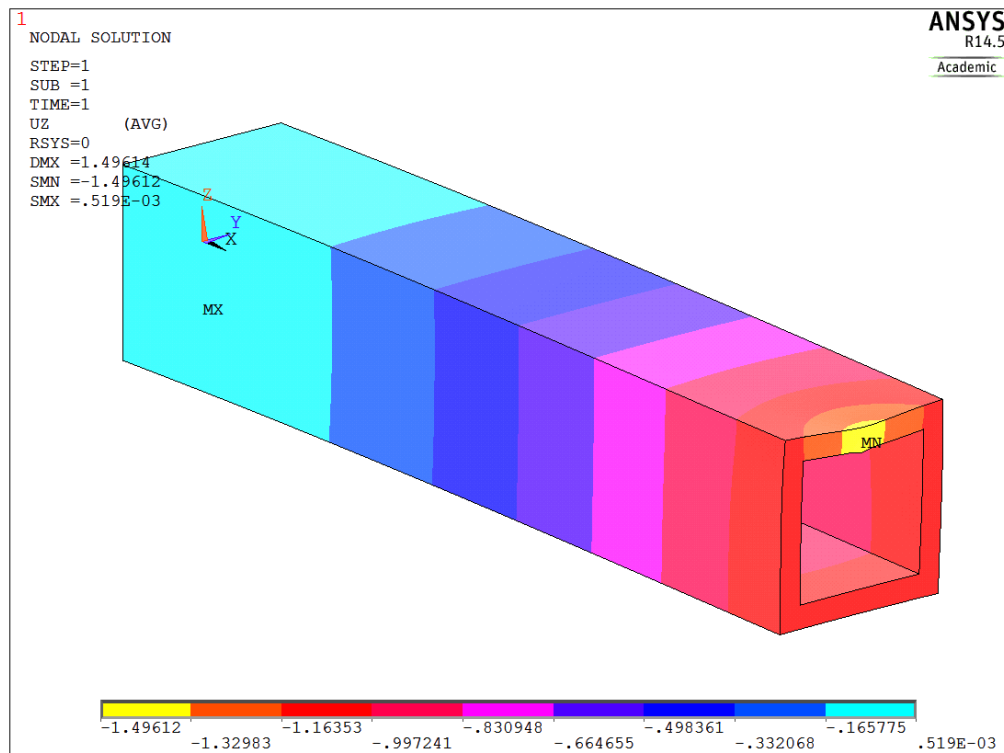


Figure 5.13: 3D reference mesh model in ANSYS

also well-known the poor performance of the tetrahedrons for capturing bending strains.

The deformation in the upper flange is higher in the meshfree model by around $\Delta U_{y-top} \approx 0.2$ for both linear and nonlinear formulations. In the nonlinear case, the upper corner nodes of the coarse meshfree grid are influenced by the point load (Figure 5.14). Although it's a slight effect, the end cross-section artificially deforms out of its plane. This is due to the large domain of influence needed by this method in order to generate well conditioned matrices for computing the shape functions.

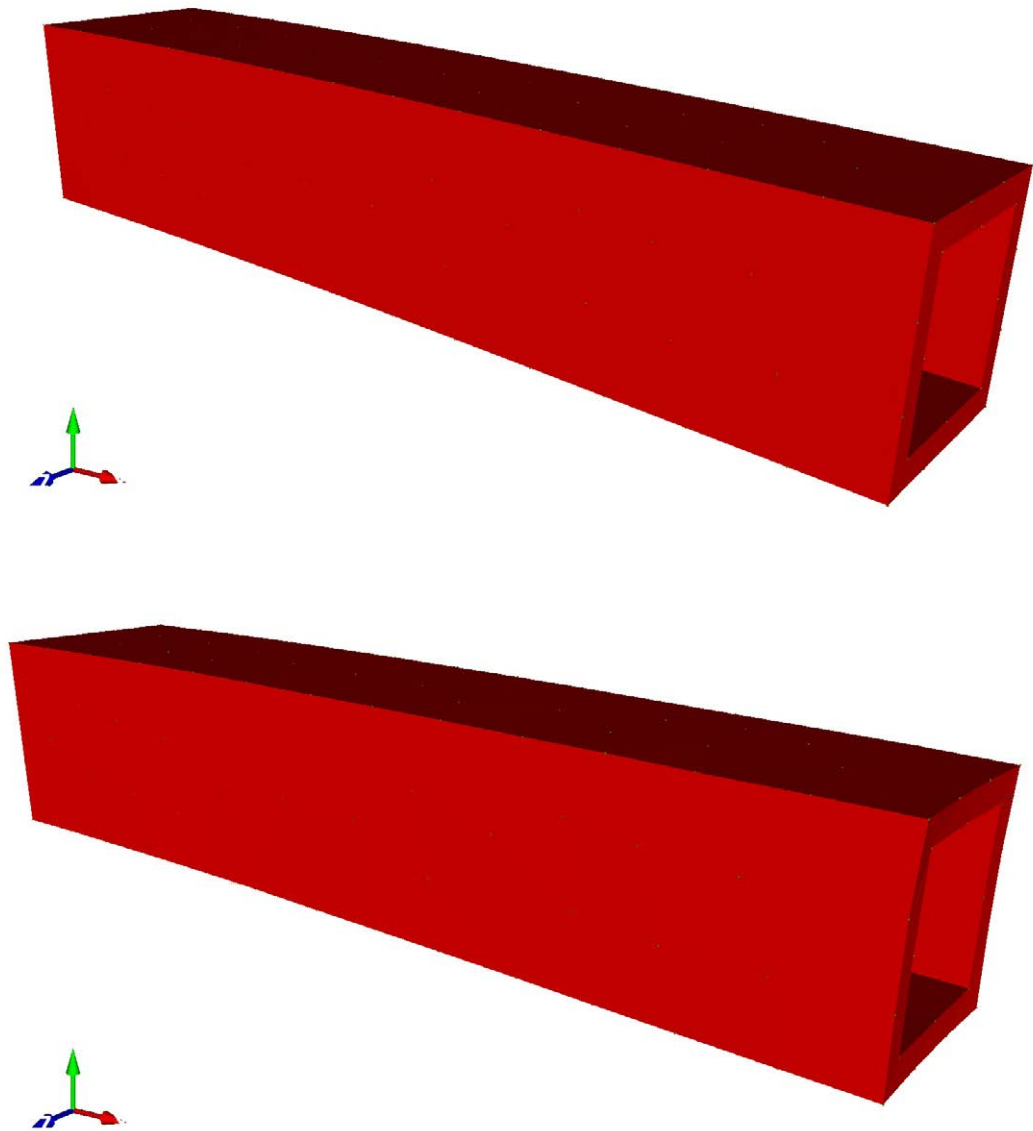


Figure 5.14: 3D coarse nonlinear meshfree solution: linear (top), nonlinear (bottom)

B. 2D solid and hollowed beams

Both the geometry and the loading conditions are not well suited at all for 2D simplified models. Nevertheless, this is a usual problem in engineering when a simplified initial calculation is made in order to evaluate a concept. For these 2D models, a plane strain formulation will be used. The load needs to be scaled by the thickness of the beam that is being modelled, leading to a 2D load of magnitude $F_{2D} = 1$

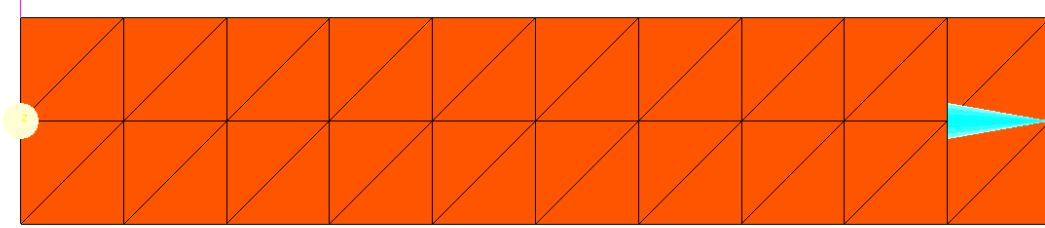


Figure 5.15: 2D solid coarse model

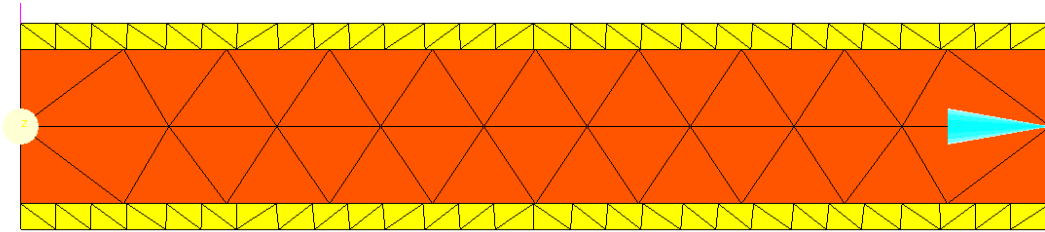


Figure 5.16: 2D composite grid model

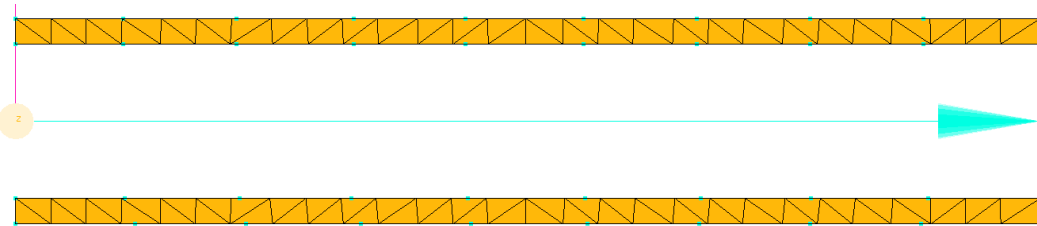


Figure 5.17: 2D hollowed grid model

Three different models have been created in order to approximately solve the problem:

- 2D solid coarse model (Figure 5.15). Represents the usual approach using a continuous model and neglecting the hollowed section. This mesh is solved using FEM approximation. Maximum displacement of the tip is $U_{y-top} = -0.4$ and $U_{y-top} = -0.35$, for the linear and nonlinear formulations. As expected, it is far stiffer than the real behaviour of the 3D beam.



Figure 5.18: 2D solid coarse model deformation

- 2D composite grid model (Figure 5.16). It uses a coarse grid for the hollowed part and a more dense nodal distribution for each of the beam flanges. The different domains define the integration points for the meshfree method, while all the nodes will be used for building the shape functions of the GPs. Maximum deformation for the beam is $U_{y-top} = -0.41$ for the linear formulation. When the nonlinear form is used, the deformation increases instead of reducing its value, as it would be expected when membrane effects are considered. The resulting $U_{y-top} = -0.43$ comes by the better capture of the local strains—induced by the point load—in the denser grid.

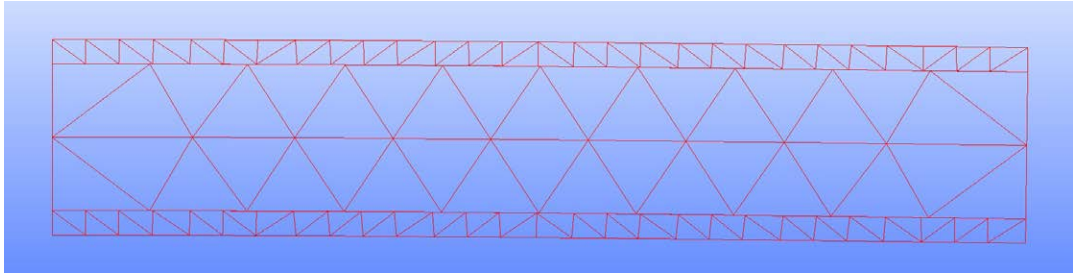


Figure 5.19: 2D composite grid deformation

- 2D hollowed grid model (Figure 5.17). The hollow material is removed, and cells containing integration points are placed only in the flanges of the actual cross-section of the 3D beam with the x-y plane. The shape functions are defined with a domain of influence that is large enough to connect both upper and lower sub-grids. As there is no material to model the side webs of the beam, the shear stress between both flanges is only transmitted by a much smaller area, thus increasing the total deformation. Also, a different behaviour has been detected between the linear and nonlinear. An average value is also calculated for the non-existing centre line end points, being the displacements at the loaded tip as follows:
 - Linear formulation: $U_{y-top} = -1.48$, $U_{y-bot} = -1.45$, with an average value of $U_{y-avg} = -1.46$.

- Nonlinear formulation: $U_{y-top} = -1.65$, $U_{y-bot} = -0.31$, with an average value of $U_{y-avg} = -0.98$.

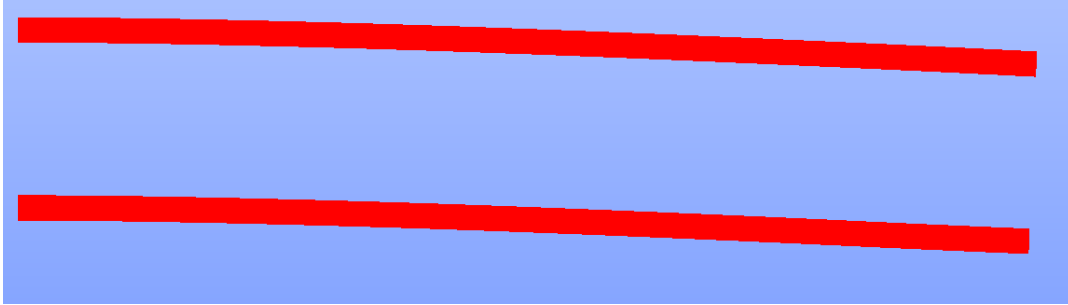


Figure 5.20: 2D hollowed grid model deformation

5.3.2 Results summary and discussion

Apart from the results presented, the Euler-Bernoulli beam model have been added to the summary in Table 5.5. To calculate the maximum deflection due to the E-B theory, the specific inertia for each model section has been applied. As the beam is not particularly slender, the E-B is not considered a reference result, but more a guide in order to see how differently the models analysed behave.

To compute the relative errors, the ANSYS 3D solid beam results are used. Depending on the case, either the top or the bottom deformation is compared. In general, the bottom value is chosen. As that lower flange is away from the singular point load, it represents the global behaviour of the beam. In the case of the the hollowed 2D model, each flange deflection is compared to the correspondent reference case flange.

It must be noted again that the hollowed 2D beam relies in the shape functions and not in any domain for transmitting the shear stress between the flanges. This is clearly method-specific and allows no generalization, so the results need to be taken with caution. In any case, the response is better than expected, and would need further investigation.

In general, meshfree shape functions provide an improvement compared to the FEM models. It can be concluded that this validation exercise shows promising results from the meshfree interpolation schemes.

Table 5.5: Results summary for loaded beam models

Model	E-B U_{max}	NL U_{max}	NL error	L U_{max}	L error
FEM ANSYS TOP	-0.77	-1.39		-1.39	
FEM ANSYS BOT	-0.77	-1.05		-1.1	
RPIM Hollowed 2D top	-1.24	-1.65	0.18	-1.48	0.06
RPIM Hollowed 2D bot	-1.24	-0.31	-0.71	-1.45	-0.32
RPIM Hollowed 2D average	-1.24	-0.98	-0.07	-1.46	-0.33
FEM Hollowed 3D	-0.77	-0.65	-0.38	-0.72	-0.35
EFG Hollowed 3D	-0.77	-0.82	-0.21	-0.93	-0.15
RPIM Solid 2D	-0.39	-0.35	-0.66	-0.4	-0.64
RPIM Solid 2D – two meshes	-0.39	-0.43	-0.59	-0.41	-0.63

One drawback of meshfree methods is that, with equal number of degrees of freedom (i.e. nodes), the cost of computation is higher to FEM. A set of three studies are presented in this chapter comparing efficiency between meshfree methods and FEM, thus trying to identify cases where their application can provide an advantage.

First section details a nonlinear very flexible double pendulum. The comparison made suggests that meshfree methods may have higher efficiency as they give more accurate solutions at a given computational cost, and has been published in [35].

6.1 Mechanical benchmark

In this section we present the analysis of a planar double pendulum as a simple example of a flexible multibody system, composed of a rigid and a deformable bar. One end of the rigid bar is connected by a revolute joint to a fixed point, and the other end is connected to a flexible beam at one point by another revolute joint, which is a rectangle of an hyperelastic material (Saint-Venant Kirchhoff) with a 2-D plane strain formulation.

The configuration of the system is parametrized by a set of *inertial cartesian coordinates* of selected points of the mechanism. In this case, these points are the two extremes of the rigid bar and the nodes associated to the spatial discretization of the flexible beam.

The rigid bar is defined by the cartesian coordinates of the nodes at both ends and a constant distance constraint between them. The inertia properties of the rigid

bar is completely defined by its mass matrix, given by:

$$\mathbf{M}_{bar} = \begin{pmatrix} m/3 & 0 & m/6 & 0 \\ 0 & m/3 & 0 & m/6 \\ m/6 & 0 & m/3 & 0 \\ 0 & m/6 & 0 & m/3 \end{pmatrix}, \quad (6.1)$$

m being the total mass of the bar. Denoting by $\mathbf{q}_i, \mathbf{q}_j$ the coordinate vectors of the end points of the bar, the constant distance constraint between them is given by:

$$\Phi = \|\mathbf{q}_j(t) - \mathbf{q}_i(t)\|^2 - l_0^2 = 0, \quad (6.2)$$

l_0 being the length of the rigid bar. The constraints associated to the revolute joint on the fixed end of the bar and the joint between the rigid bar and the flexible beam are formally the same. Both express that the distance between two points i, j is null:

$$\Phi = \mathbf{q}_j(t) - \mathbf{q}_i(t) = \mathbf{0} \quad (6.3)$$

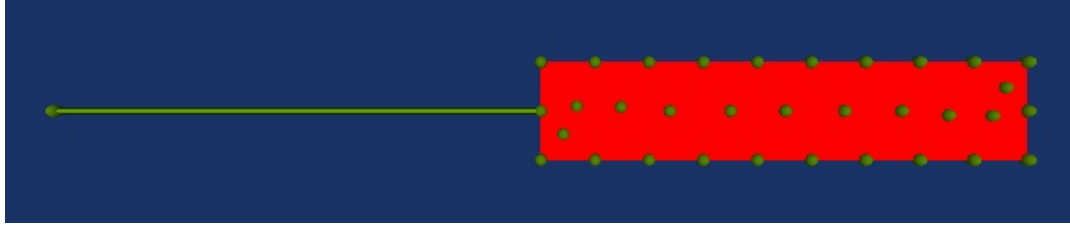
The Finite Element Method (FEM) is compared to the Meshfree formulation using Multiquadrics Radial Basis Functions (RBF), with no polynomial terms. The FEM mesh is used in this case to define the integration cells, and in both cases the equations of motion are integrated using the same implicit integrator.

Two different initial conditions are considered in order to produce qualitatively different deformations, at least during the first seconds of the simulation (see Figure 6.1).

- In Case I both bars are horizontal. The movement is simulated for a full period and the deformation of the flexible bar during the first seconds of the simulation is mainly axial.
- In Case II the rigid bar is vertical while the flexible bar is horizontal, producing a high bending deformation at the beginning of the motion. The simulation is shorter than in the previous case because we are interested in analyzing the lash-type behavior that takes place during the first quarter period.

In both cases the bars are released from rest from their initial positions, and three different elastic modulus are considered for the elastic bar. As this value gets smaller, the deformations are higher, causing larger distortions of the FEM mesh affecting the performance of the formulations. Table 6.1 details the parameters of the both cases.

Different meshes are used for modelling the flexible beam. The number of degrees of freedom of the coarser mesh is 64, and consecutive refinements maintain



a) Case I



b) Case II

Figure 6.1: Initial configurations

coincident nodes allowing to directly compare the variables of interest at the same locations. The simplicity of the geometry has made possible to define very regular meshes in all cases.

6.1.1 Error estimation

The error is estimated using an energy norm [64] of the state vector $\mathbf{z}(t) = \{\mathbf{q}(t), \dot{\mathbf{q}}(t)\}^T$:

$$\begin{aligned} \|\cdot\|_E : \mathbb{R}^{2n_{dof}} &\longrightarrow [0, \infty) \\ \mathbf{z}(t) &\longrightarrow \|\mathbf{z}(t)\|_E := \frac{1}{2} \dot{\mathbf{q}}(t) \cdot \mathbf{M} \dot{\mathbf{q}}(t) + \frac{1}{2} \mathbf{q}(t) \cdot \mathbf{K} \mathbf{q}(t) \end{aligned} \quad (6.4)$$

The absolute configuration error at coincident nodes is computed at each step as:

$$\bar{\mathbf{z}}(t_n) = \mathbf{z}(t_n) - \mathbf{z}_{ref}(t_n) = \{\bar{\mathbf{q}}(t_n), \bar{\dot{\mathbf{q}}}(t_n)\}^T \quad (6.5)$$

where:

$\mathbf{z}(t_n)$ is the computed state vector at t_n

$\mathbf{z}_{ref}(t_n)$ is the reference solution for the state vector at t_n .

The reference solution has been computed with a 1433 node FEM model (2866 degrees of freedom). Note that $\mathbf{z}(t_n)$ includes only the subset of DOFs of the nodes that

Table 6.1: Details of the numerical experiment

	Properties
Rigid bar	$l_{0\text{CaseI}} = 40 \text{ m}$, $l_{0\text{CaseII}} = 20 \text{ m}$, $m = 4 \cdot 10^{-2} \text{ kg}$
Flexible bar	Dimensions: $40 \times 4 \text{ m}$ FEM: CST elements (3 nodes, 1 GP) Meshfree: 1 GP per cell, MQ shape functions ($\alpha_c = 3.5$, $q = 1.03$)
Flexible material	SVK model Density, $\rho = 0.2 \text{ kg/m}^3$ Poisson coefficient, $\nu = 0.3$ Young modulus, variable, $E = (3000, 4000, 5000) \text{ Pa}$
Constraints	Penalty method penalty parameter $\alpha = 10^6$
Time integration	$t_0 = 0 \text{ s}$, $t_{f\text{CaseI}} = 18 \text{ s}$, $t_{f\text{CaseII}} = 4.5 \text{ s}$ Trapezoidal rule, $\Delta t = 0.05 \text{ s}$

exist in both meshes, as the reference solution is computed with a finer mesh.

Finally, the absolute energy error in t_n is computed as:

$$\|\bar{\mathbf{z}}(t_n)\|_E := \frac{1}{2} \bar{\mathbf{q}}(t_n) \cdot \mathbf{M} \bar{\mathbf{q}}(t_n) + \frac{1}{2} \bar{\mathbf{q}}(t_n) \cdot \mathbf{K}_0 \bar{\mathbf{q}}(t_n) \quad (6.6)$$

\mathbf{M} being the constant mass matrix of the system and \mathbf{K}_0 the initial stiffness matrix.

6.1.2 Results

The solution has been obtained using three different mesh densities (33, 107 and 381 nodes) and three different Young modulus' values (3000, 4000 and 5000 Pa) in order to observe the effects of flexibility on the quality of the solution.

For values of the elastic modulus lower than 3000 Pa, FEM simulations did not converge due to extremely large distortions of the mesh. On the other hand, the meshfree method showed higher robustness since convergence was achieved in all cases down to $E = 500 \text{ Pa}$. Based on these results, and taking into account that the reference solution has been obtained with FEM, the lowest modulus considered in both cases has been 3000 Pa.

In spite of the simulation code is not optimized nor parallelized, its object oriented design makes both methods share the maximum number of functions, so computational times can be compared.

A. Results of Case I

The pendulum departs from rest from an horizontal position and the motion is computed up to an instant where the rigid bar almost reaches again the initial configuration. The first half of the movement is relatively smooth, while the second half shows larger oscillations. The computations using different discretizations does

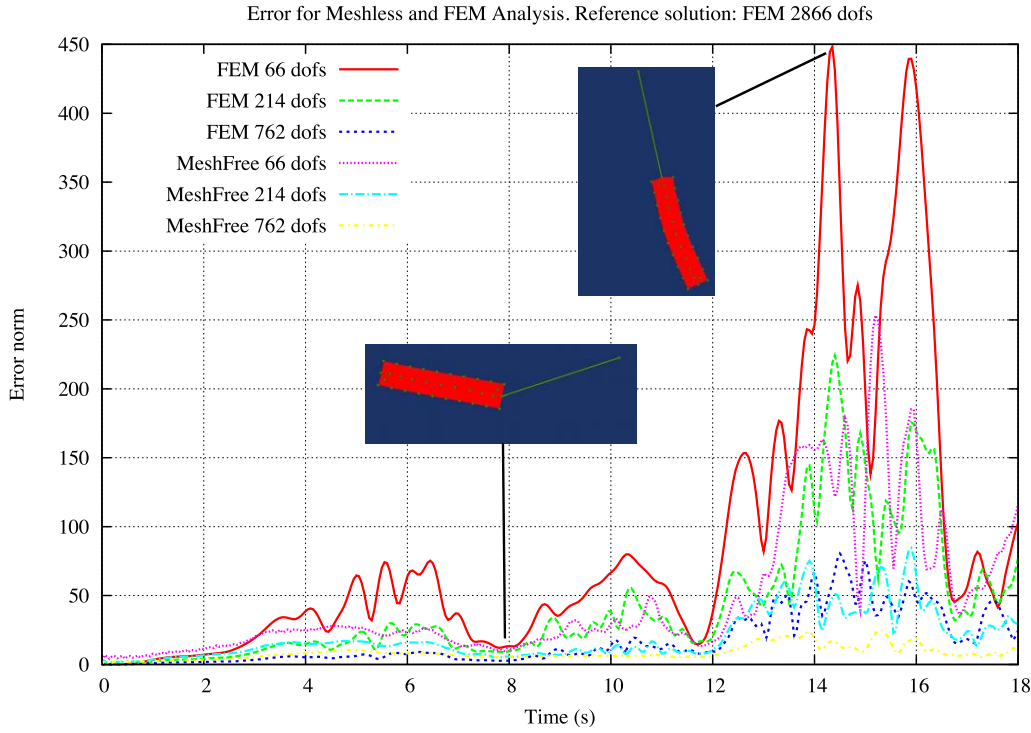


Figure 6.2: Case I: absolute error vs. time. $E = 5000$ Pa

not show a chaotic behavior and, therefore, results can be confidently compared to the reference solution for error estimation.

The evolution in time of the error for each discretization is shown for the maximum and minimum values of the elastic modulus in Figures 6.2 and 6.3 respectively. A first qualitative analysis of the results shows a relation between the error value and the magnitude of the deformation, which is related to the mesh distortion. As the figures suggest, the maximum error corresponds to a highly tensioned bar, and the error is low when the beam is less deformed. These results indicate that the meshfree method is less sensitive to the strain magnitude than FEM.

Figure 6.4 shows the dependency of the maximum absolute error with the number of degrees of freedom. We can observe that both methods converge to the reference solution, but the meshfree method has higher accuracy at equal number of degrees of freedom and, it has also a slightly higher convergence rate.

In order to explore a potential advantage of the meshfree method in terms of efficiency, Figure 6.5 shows the relation between the maximum error and the total computing time. It becomes apparent that the meshfree method takes less computing time to achieve a given accuracy in spite of the fact that the computation of the local matrices takes more time, because the quadrature points have more number of support nodes compared with FEM.

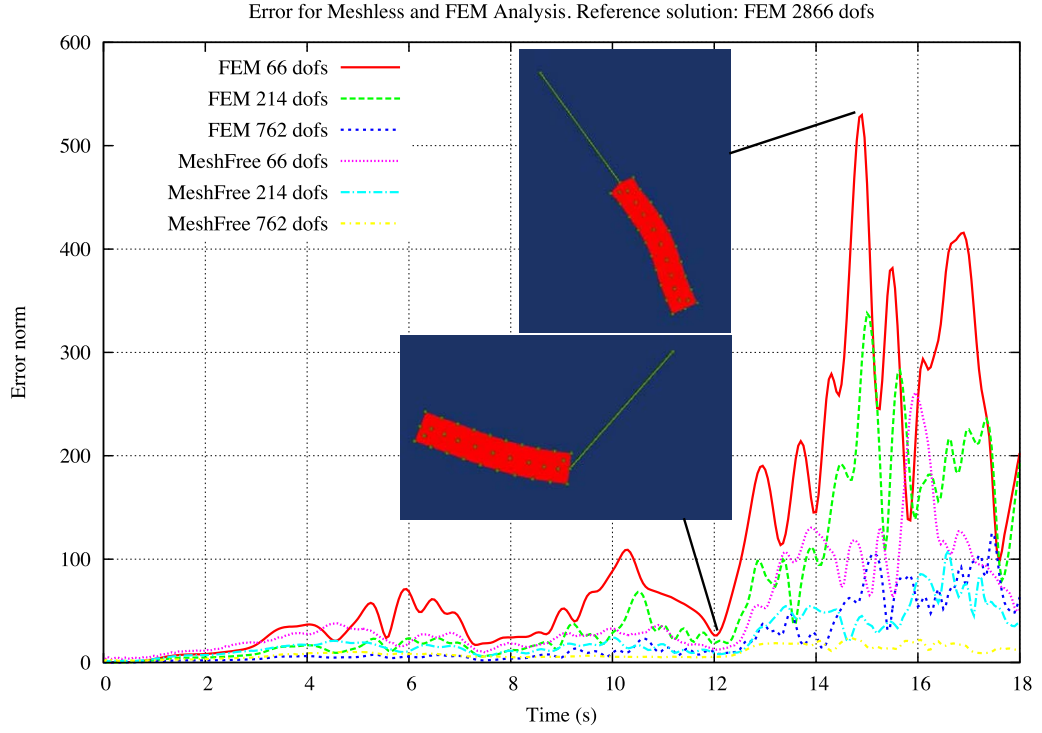


Figure 6.3: Case I: absolute error vs. time. $E = 3000$ Pa

B. Results of Case II

As described previously in this section, the computed motion is shorter than in Case I, and the initial conditions are chosen such that the bending deformation is dominant during the first seconds of the motion. This situation clearly differs from Case I, where the longitudinal deformation of the beam was more important.

The evolution in time of the absolute error computed with both methods is different. The error of the FEM model oscillates following quite closely the bending of the flexible bar; but the longitudinal deformation seems to have more influence in the behaviour of the meshfree model. For $E=5000$ (Figure 6.6) the maximum error for the meshfree method occurs in $t = 4.2$ s, when the main effect is the longitudinal deformation of the beam due to the inertia and the force exerted by the rigid bar (this effect is not visible in Figure 6.6, but it can be clearly identified in an animation of the motion). When the value of the Young modulus goes down to 3000 Pa (Figure 6.7), the maximum deformation has both effects (bending and longitudinal deformation) and the peaks of error are closer but they still do not occur in the same moment. Again, the maximum errors occur when the strains are larger and the FEM mesh is more distorted, and the results indicate that the meshfree method is less sensitive to the strain magnitude.

In terms of convergence, the meshfree discretization shows in this case higher

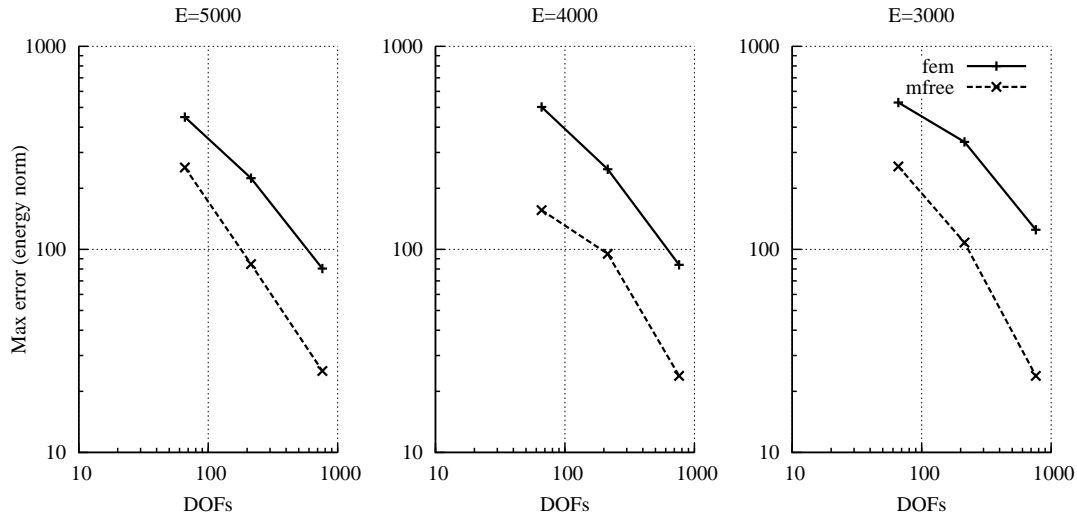


Figure 6.4: Case I: Maximum error vs. number of degrees of freedom.

accuracy for equal number of degrees of freedom (Figure 6.8), as it happened for Case I, but in this case the convergence rate is lower than for FEM.

Comparing to clock time instead of number of degrees of freedom, the advantage of the meshfree method only shows up for the coarser meshes (see Figure 6.9). This behaviour suggests that the efficiency of the meshfree method gets better compared to the FEM when the beam gets more flexible and the deformations and the mesh distortions are higher. The computations with the three different values of the elastic modulus show this trend.

Figure 6.10 shows the evolution of the accumulated computing time for the first ten steps. These results confirm that the preparation of the internal data structures and the construction of the tangent matrix significantly penalize the computing time of the meshfree method compared with FEM. Firstly, because the local matrices are bigger, and secondly and more important, because the computation of the shape functions implies the solution of a linear system for each quadrature point. The y-axis of Figure 6.10 is shown in log scale in order to make more visible the initial delay that these effects produce at the beginning of the computation. It is apparent that, for the same number of degrees of freedom, the meshfree model takes more time than FEM to start the computation. Nevertheless, it must be recalled that the use of the total Lagrangian formulation demands the shape function computation to be performed just once at the beginning. Note also that this delay is relatively small compared to the time that it takes to compute the whole simulation.

Finally, Figure 6.11 presents an example of the robustness of the meshfree method when dealing with an extreme deformation, corresponding to Case II with an elastic modulus $E = 500$ Pa, which is ten times less than the original. No error analysis or

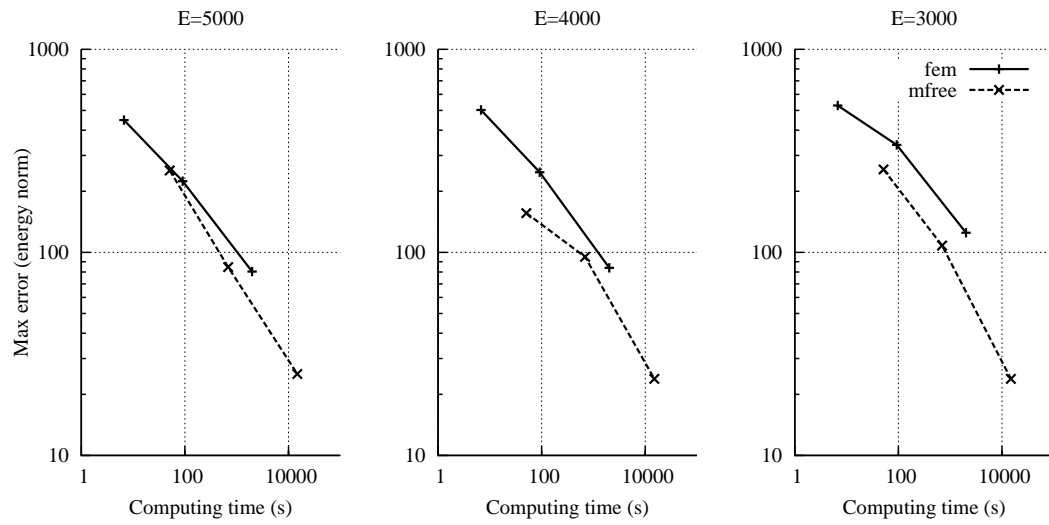


Figure 6.5: Case I: Maximum error vs. computing time

comparison could be made in this occasion since convergence was not achieved with the FEM model.

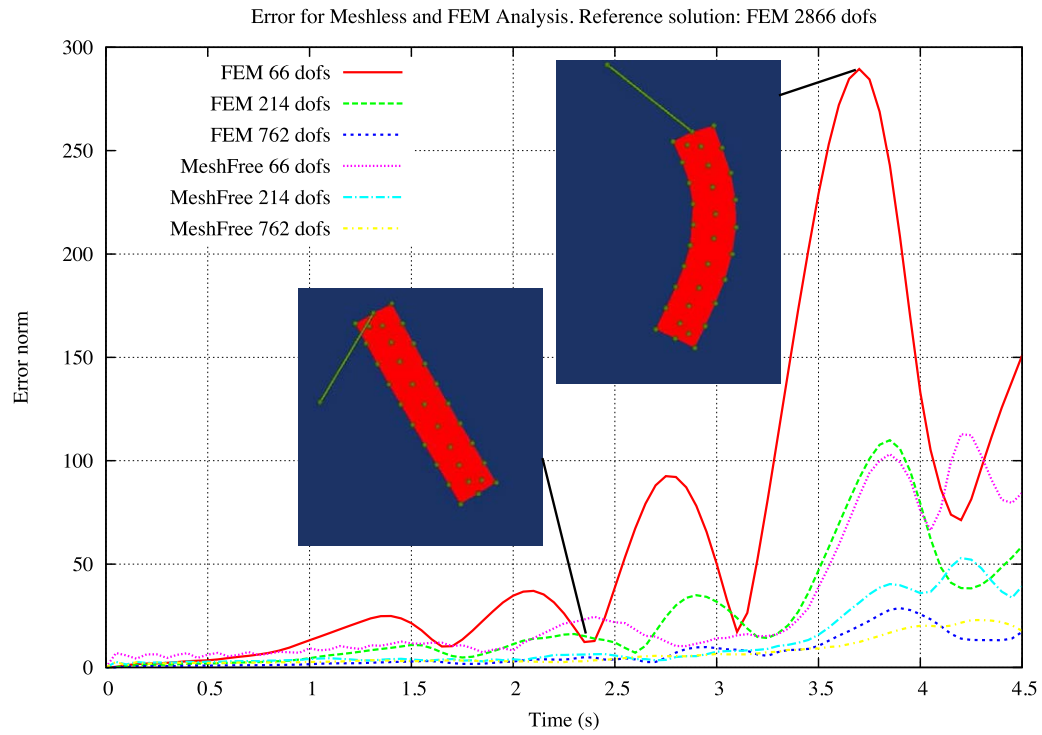


Figure 6.6: Case II: absolute error vs. time. $E = 5000$ Pa

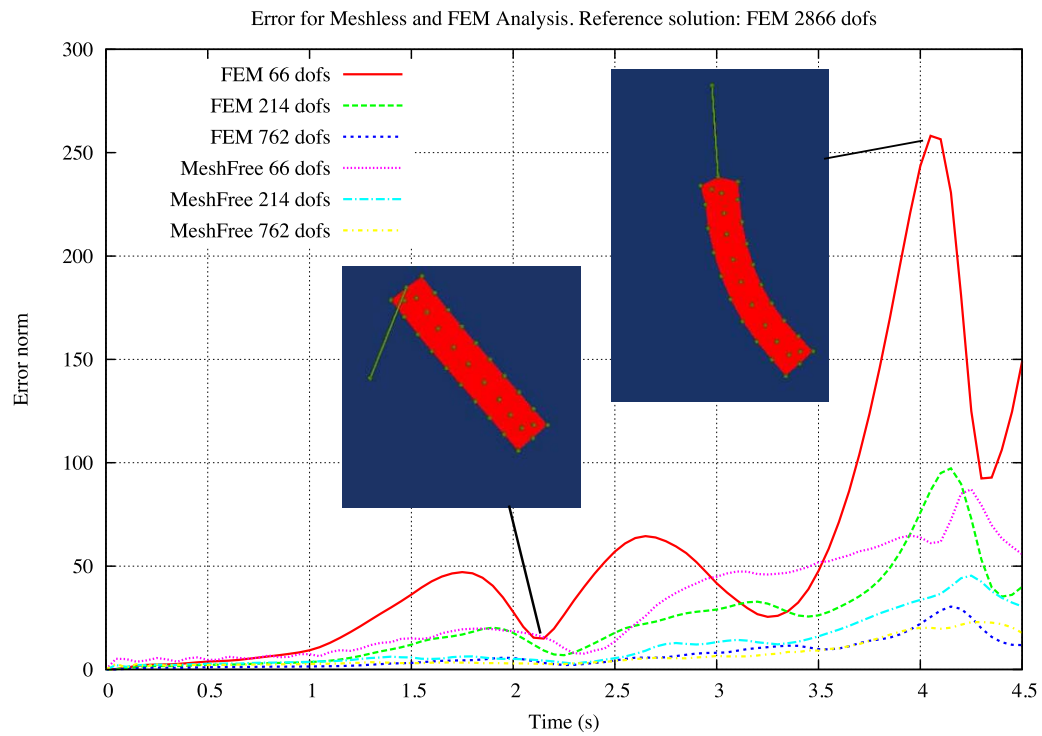


Figure 6.7: Case II: absolute error vs. time. $E = 3000$ Pa

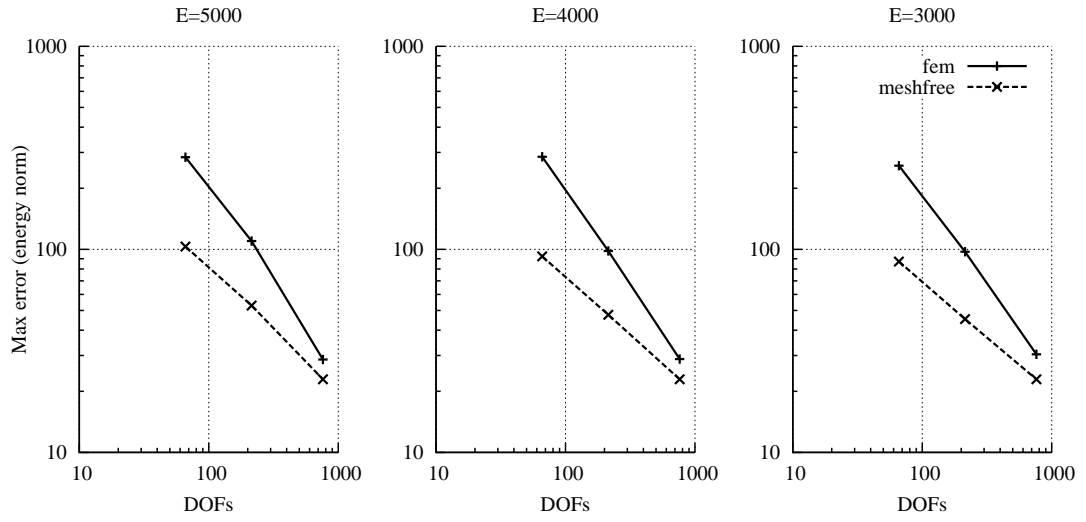


Figure 6.8: Case II: Maximum error vs. number of degrees of freedom

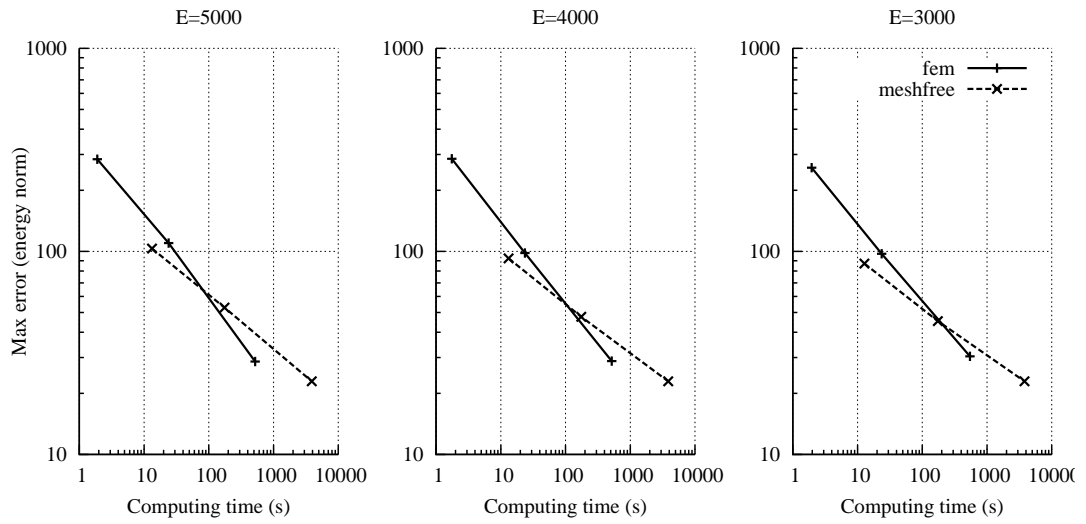


Figure 6.9: Case II: Maximum error vs. computing time

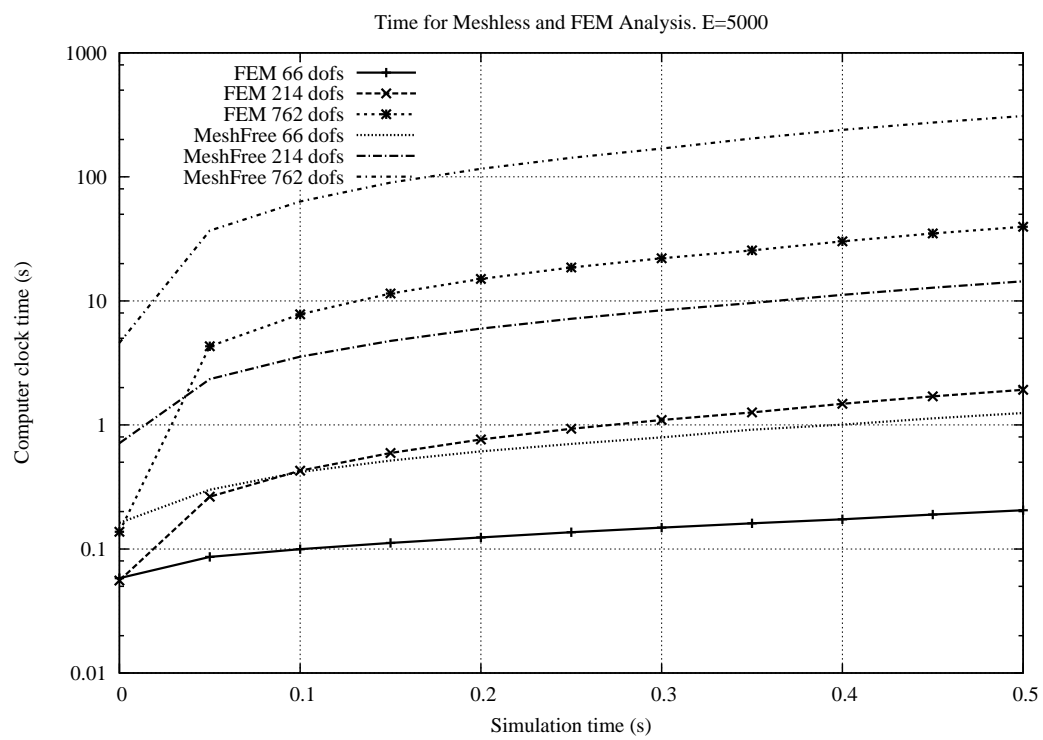


Figure 6.10: Case II: Computing timer vs. simulation time. Detail of the ten first time steps

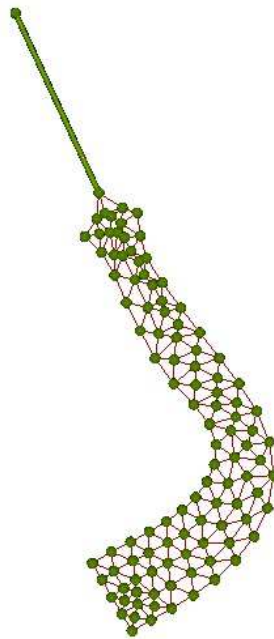


Figure 6.11: Case II. System configuration at $t = 4.05$ s for $E = 500$ Pa (meshfree, 107 nodes). The connectivity among nodes define the integration cells

6.2 Thermal benchmark

6.2.1 IFMIF LIPAc slits beam model

The IFMIF ion beam will be have the highest current beam ever achieve in a continuous wave (CW) accereratpr. The heat power density can be as high as 5.8 GW/m² for some diagnostics. The beam parameters in the diagnostic plate are the following:

- 1.125 MW
- D⁺ 5 MeV (MEBT), 9 MeV (HEBT)
- $I = 125$ mA
- Duty cycle from 0.1% to 100% (CW), $f = 1$ Hz.

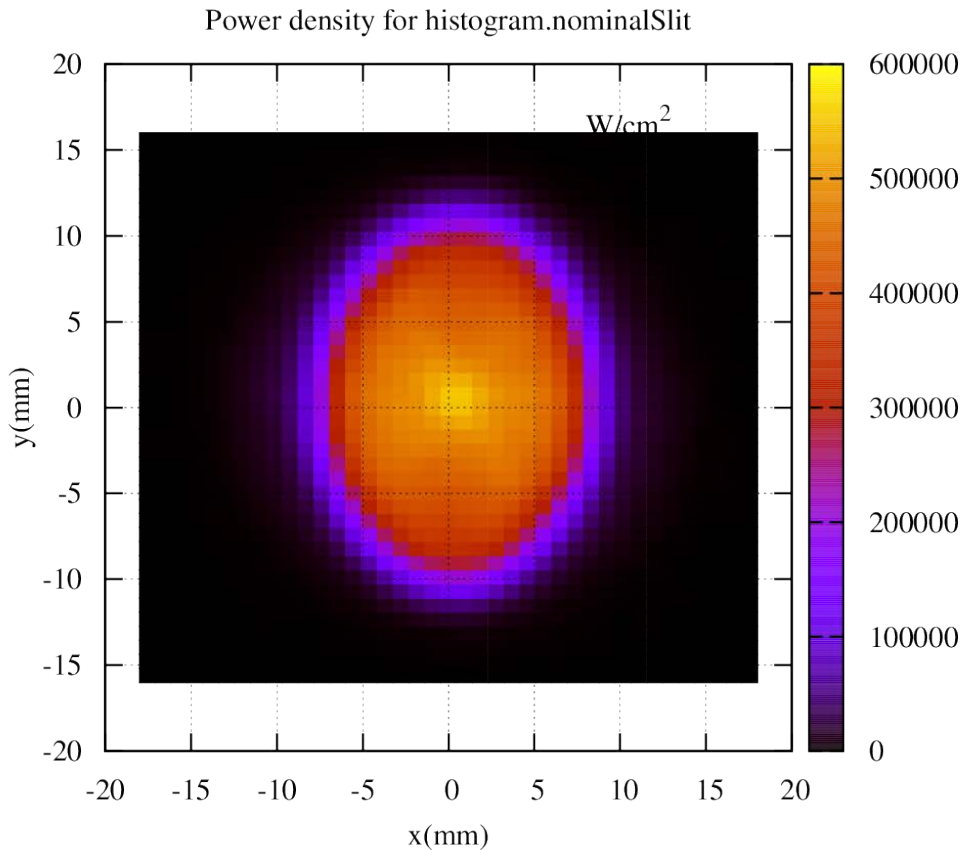


Figure 6.12: Beam power footprint at the diagnostic plate

LIPAc ions penetrate just a hundred of microns in the solid material [35]. The input from Beam Dynamics for the analysis of any device consists mainly on the beam

properties (footprint and divergence) at a section of the line close to the device. This information gives directly the heat flux $\mathbf{q}(\mathbf{x})$ at any spatial point. Considering beam moves along the z -axis and has a bi-Gaussian shape:

$$\mathbf{q}(\mathbf{x}) = \frac{P \cdot \exp \left\{ -\frac{(x-x_0)^2}{2(\sigma_x + \sigma'_x(z-z_0))^2} - \frac{(y-y_0)^2}{2(\sigma_y + \sigma'_y(z-z_0))^2} \right\}}{2\pi(\sigma_x + \sigma'_x(z-z_0))(\sigma_y + \sigma'_y(z-z_0))} \mathbf{e}_z \quad (6.7)$$

If long pulses or CW behaviour is simulated, the load can be applied as a Neumann boundary condition using Eq. (6.7) as the normal vector at any point of the boundary $\mathbf{n}(\mathbf{x})$ can be calculated using the position of its neighbours along the boundary. These neighbors can be found by the node search algorithm used for the shape function calculation or by a predefined connectivity, which is always needed if the body is non-convex.

For short pulses, the volumetric power density must be used. An ionization table, $\mathbf{\Xi}(a, x_n)$, is prepared using SRIM code results in the beam facing wall's (BFW) normal direction depending on the incidence angle, a . The heat rate value at the integration points is calculated as shown in Figure 6.14. For a Gauss point j , the rate is averaged from the contributions of its support nodes i , provided that the support size radius is much greater than the ionization depth:

$$r_j = \sum_{i=1}^{SP} \frac{r_{ij}}{SP} = \frac{1}{SP} \sum_{i=1}^{SP} \frac{q_i}{E} \mathbf{\Xi}(a_i, (\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_i) \quad (6.8)$$

A great advantage of this approach is that the beam must only be defined in a plane. As the body moves and deforms, the simulation procedure computes and updates the heat rate for each time step.

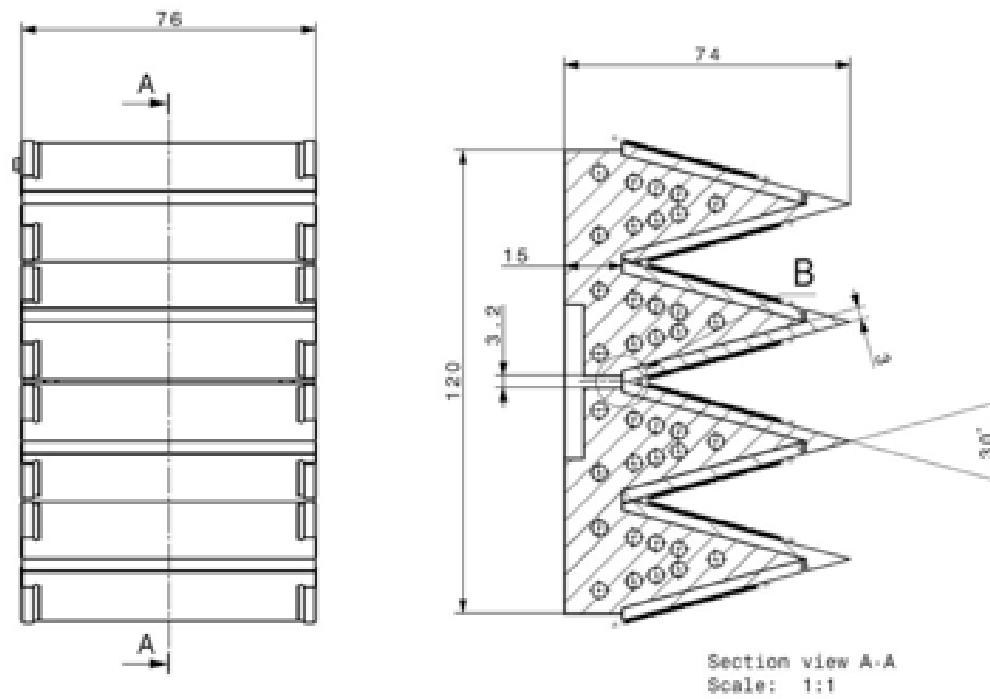


Figure 6.13: Design of the LIPAc interceptive slit system

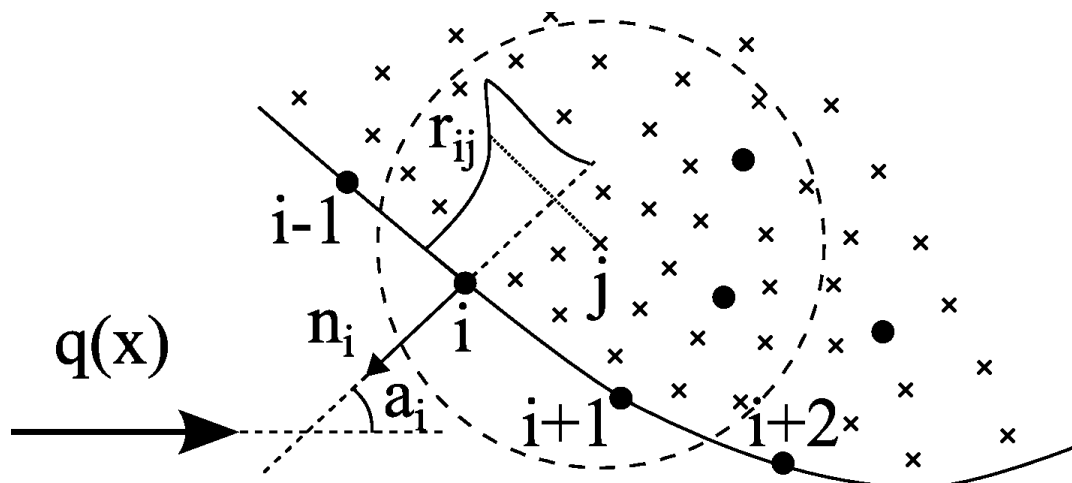


Figure 6.14: Computing of volumetric heat rate at an integration point

6.2.2 IFMIF LIPAc slits section analysis

The method has been used for the simulation of the slits diagnostic of Figure 6.15 with the following beam parameters: $P = 1.125$ MW, $E = 9$ MeV, $\sigma_x = 4.985$ mm, $\sigma_y = 6.174$ mm, $a_0 = 15$ deg. Preliminary thermal results of a power deposition section strip show a consistent higher precision and accuracy of the solution using meshfree methods (Figure 6.18). Future work will focus on the thermo-mechanical behaviour of the BFW plate.

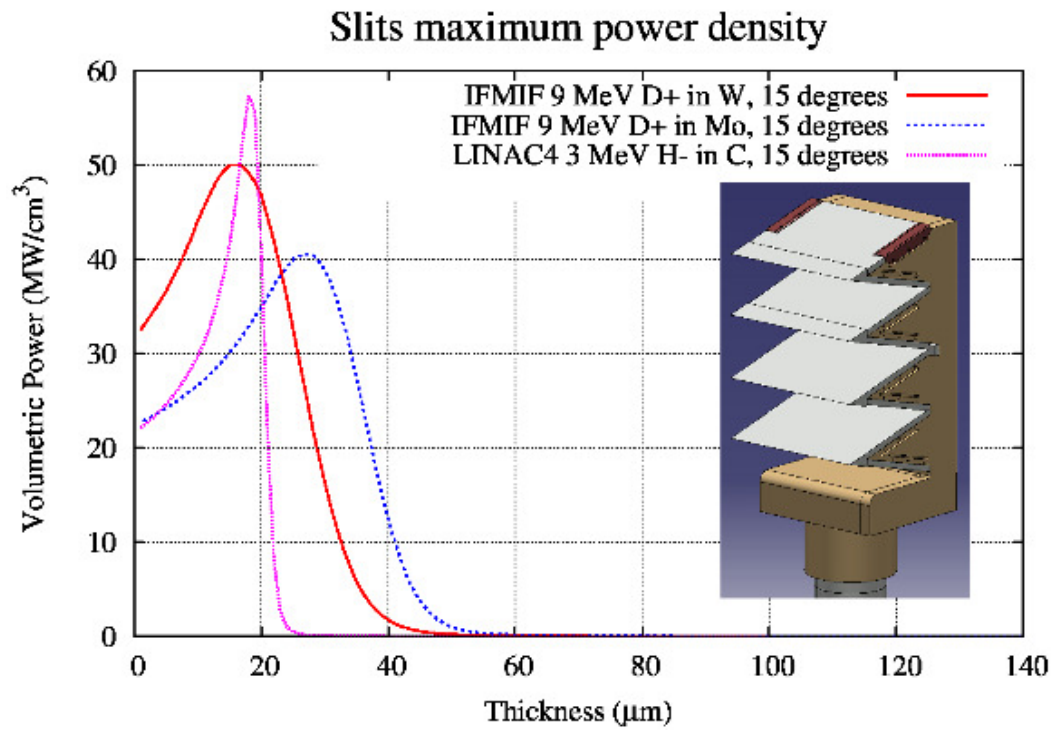


Figure 6.15: Slits design with volumetric power density loading

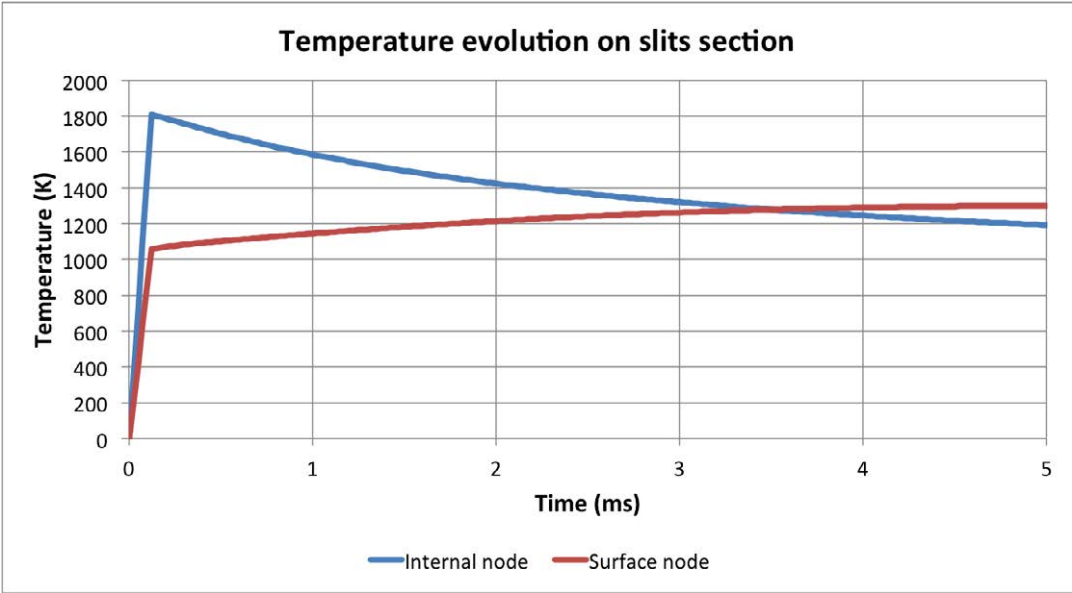


Figure 6.16: Temperature evolution of slit during beam pulse

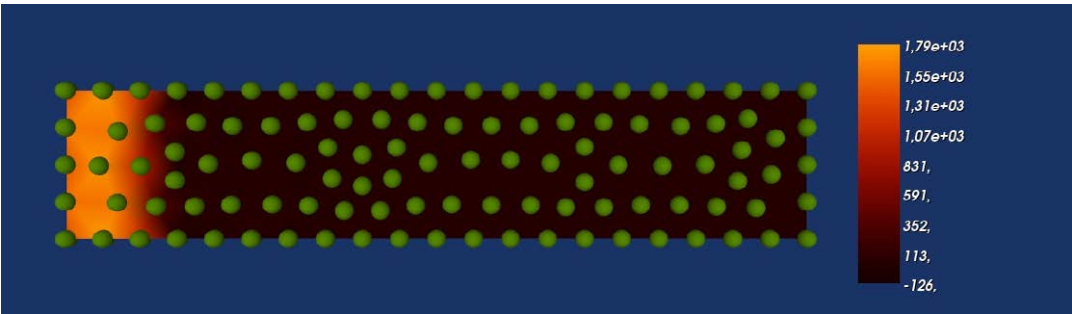


Figure 6.17: BFW Section temperatures during pulse

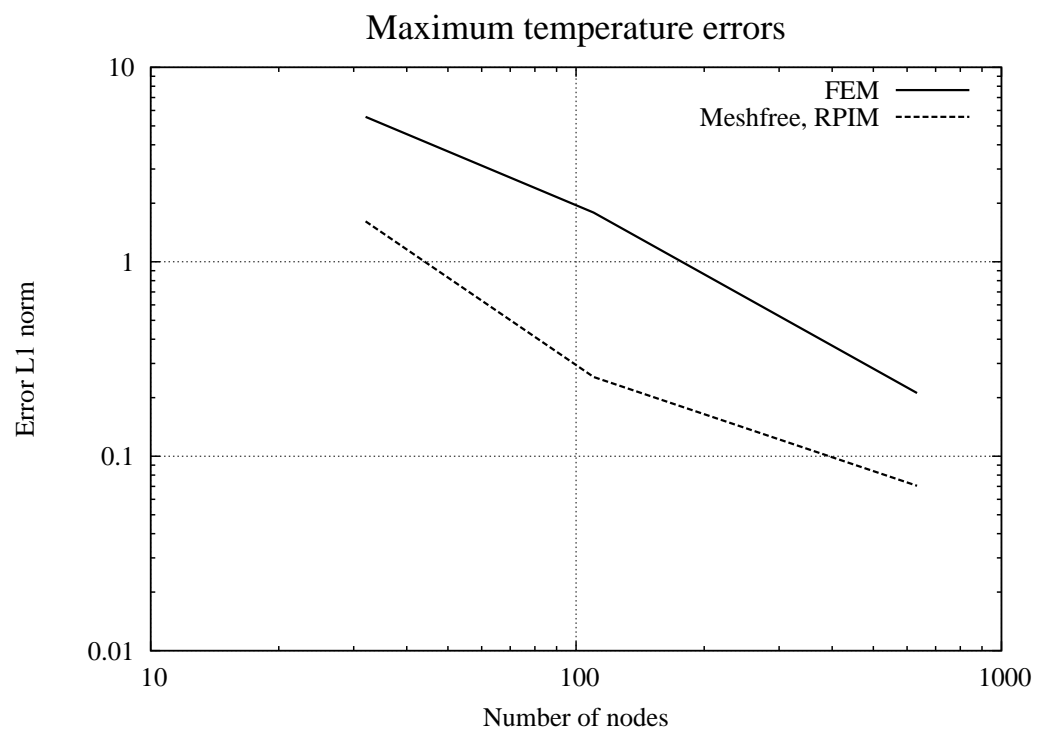


Figure 6.18: Temperatures L1 error norm comparison

6.3 Thermo-mechanical coupling benchmark

The previous two examples are mixed to compare if the results are consistent for the coupled nonlinear formulation. The double pendulum is set at t_0 in horizontal position, and loaded by a volumetric heat flux similar to the one of the previous section. Some load scaling is applied in order to use the same geometry and models as in the mechanical benchmark.

The rigid bar is located between $(-40, 0)$ and $(0, 0)$. The rectangular beam corners are $(0, -4)$ and $(40, 4)$. Finally there is a revolute joint at each end of the rigid bar, connecting the bodies between them, and the rigid bar to ground.

The reference solution is calculated using the 1532 nodes FEM mesh shown in Figure 6.19. Mechanical loading is limited to gravitational effects in vertical direction ($g = (0, -10)$), and the material properties used are the following:

- Thermal conductivity: $\kappa = 300$
- Specific heat capacity: $c_p = 1130$
- Thermal expansion: $\beta = 1\text{E-}4$
- Density: $\rho = 7850$
- Young's modulus: $E = 2\text{E}2$
- Poisson coefficient: $\nu = 0.3$



Figure 6.19: Initial configuration of the thermomechanical system

The heat load is applied for the first $1/8$ length (left hand side) of the beam, with the profile shown in figure 6.20. The heat flow is applied instantly from the start ($t = 0$), and kept constant for two steps ($t = 0.1$). After that the load is removed and the solid cools down by internal conduction until it reaches equilibrium at the end of the simulation ($t = 30$).

$$Q_{ext} = \begin{cases} 2000(40 - x) & 0 \leq x \leq 5, 0 \leq t \leq 0.1 \\ 0 & \text{otherwise} \end{cases}$$

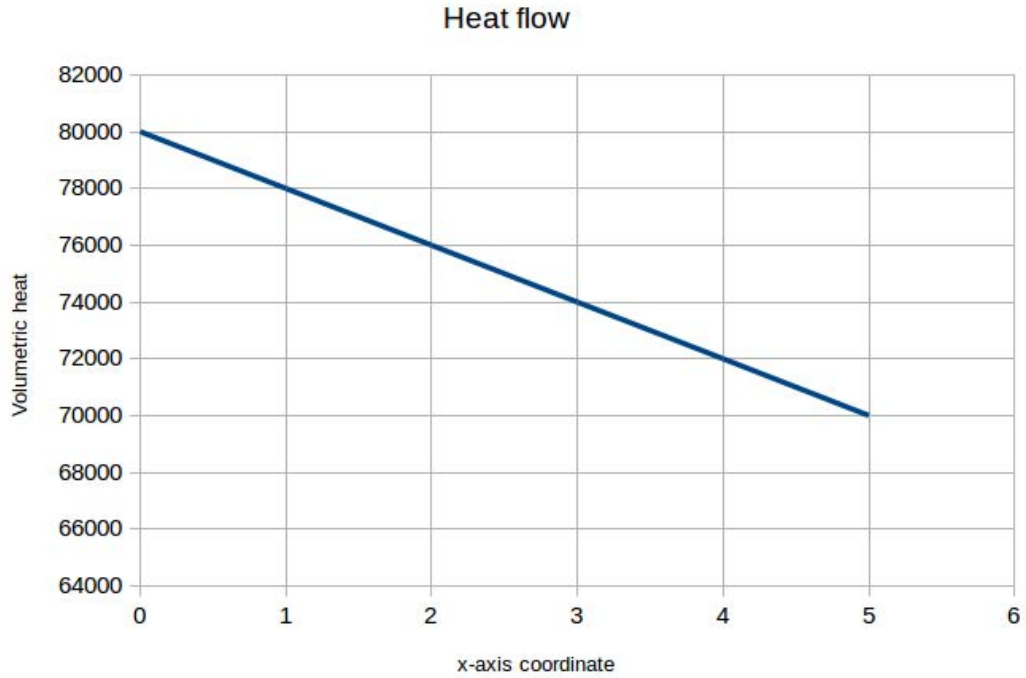


Figure 6.20: Heat flow profile during time $t = [0 : 0.1]$

6.3.1 Overall models and simulation description

The rigid bar is modelled with a penalty factor of $\alpha = 1\text{E}3$ for all the models. A more stiff behaviour is given to the joints, where the penalty factor raises to $\alpha = 1\text{E}8$.

The integration is carried out by a first order BDF (Euler implicit) integration scheme. Step size chosen sits in between the needs of the thermal load, and the gravitational deformation response time scales, $\Delta t = 0.05$. The simulation is run from $t = 0$ to $t = 30$, point where the conduction effects have led to an almost uniform temperature within the domain. The total simulation therefore requires the solution of 601 steps, considering also the analysis of the initial acceleration and heat fluxes at $t = 0$.

The 32 and 110 nodes models from the mechanical benchmark of section 6.1 are used also for this study. Figure 6.21 shows a snapshot of the simulated deformation of the reference model with 1532 nodes.

Both meshfree shape functions have been applied, and also a variation on their parameters is studied to study the effects of the size of the influence domain and the integration order. In the case of the EFG, the method shows robust enough to run a varied set of parameters for the whole simulation.

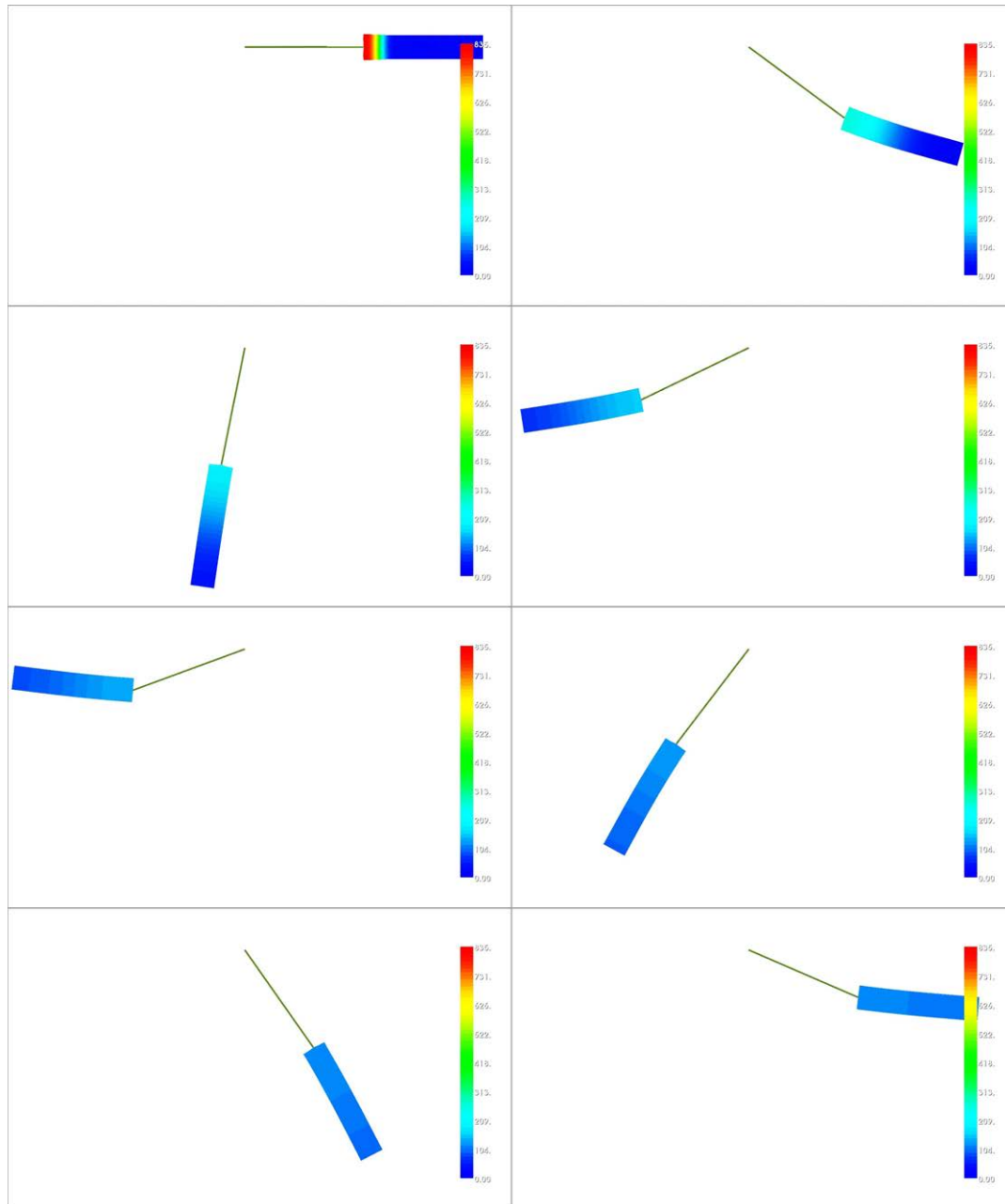


Figure 6.21: Evolution of the system configuration for the reference case, including temperature map. From top to bottom, left to right: $t = (0.01, 2.5, 5.0, 7.5, 10.0, 12.5, 15.0, 17.5)$

For the RPIM is way more difficult to achieve stability, and the simulation blows up quite easily. In fact, for the case of the 32 node model, it was not possible to achieve convergence for all steps with any parameter combination using the RBF shape functions. On the contrary, when the number of nodes where raised to 110, the method showed robust and equally accurate for different sizes of the influence domain. The lower size giving stable models as been chosen in order to optimize the computation time.

The different combinations of method parameters for the simulations are summarized in the table 6.2.

Table 6.2: Meshfree shape functions parameters

	$\alpha_i = 2.0$	$\alpha_i = 2.5$	$\alpha_i = 3.5$
1GP	EFG110	-	-
3GPs	EFG32, EFG110	RBF110	EFG32
6GPs	EFG32	RBF110	-

Important differences appear in both phases: heating and cooling. The peak temperature values are mostly determined by the volumetric load integration, and the model of the domain heating capacity (\mathbf{H} matrix). Meanwhile, the cooling differences will be dependant on the temperature gradient, and also on the conduction model (\mathbf{C} matrix).

At the same time, there is also a non-negligible elastic deformation induced by the thermal strain. This deformation is taken into account by the model both in the heating and conduction processes. In essence, the loads vary depending on the location of the GPs. As opposite to the previous benchmark case, the volumetric load profile has been chosen to be linear, in order to compare the behaviour for stepped loads with a shape that can be exactly integrated by FEM, but that have enough variation to have be influenced by the deformation process.

6.3.2 Initial heating

The peak temperature is compared between the models at time $t = 0.1$. The interpolation of the heat load is done applying the value of the flow depending on the location of the integration points, scaling it by the GP weight and cell jacobian, and then transferring that load to the support nodes using the shape functions.

The reference solution is shown in the sequence of Figure 6.22. The heat-up is almost adiabatic, and the conduction effect starts cooling down the maximum temperature zone as soon as the heat load is not applied.

To evaluate the accuracy of the heat capture, the peak temperatures at the lower left corner of the domain $((x, y) = (0, -4))$ are compared at the end of the loading

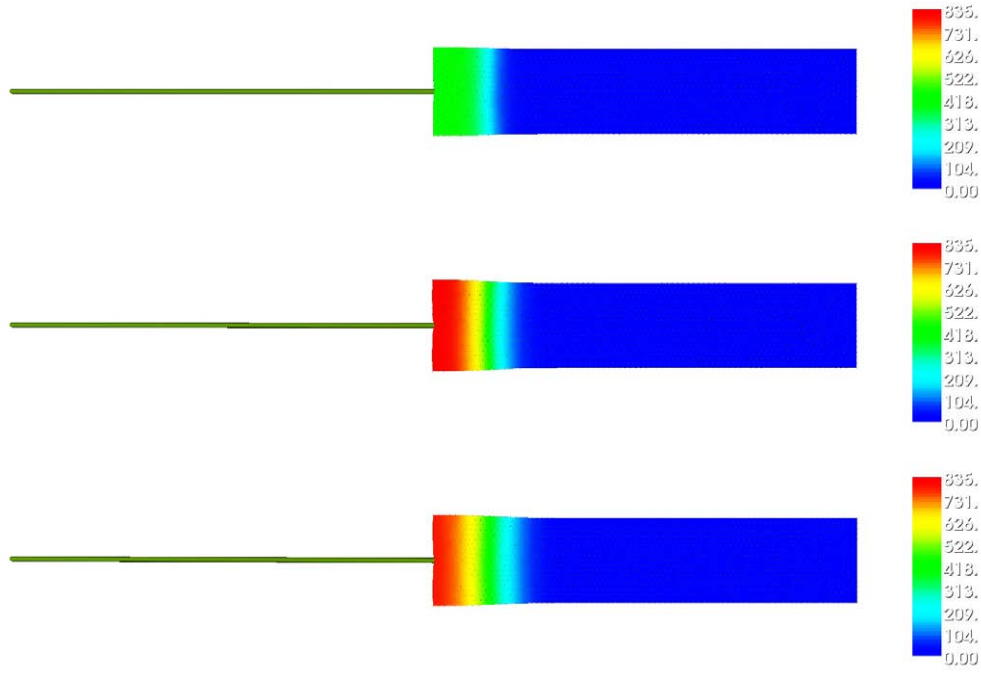


Figure 6.22: Temperature profile for the reference model for times $t = (0.05, 0.1, 0.15)$ (top to bottom)

time. Results are shown in table 6.3.

Being the load linear, the FEM models behave much better than in the thermal benchmark studied in the previous section. These present a relative low error compared to the reference solution.

Using the EFG method the relative errors are slightly higher for the 110 nodes models, and slightly lower for the 32 nodes. Increasing the number of GPs does not reduce that error. Having a high influence domain factor ($\alpha = 3.5$) reduces the accuracy of the solution, as the heat integrated by a GP inside the volumetric load will be transfer beyond it. Nodes located at $x > 5$ will receive some load. In fact, this is the effect that is affecting the accuracy of the rest of the EFG models, although with less impact, as their influence domains are smaller ($\alpha = 2.0$).

Finally, the RPIM accuracy is in the order of the EFG. This method gives higher peak temperatures than the reference solution, instead of falling short as it would be expected. The reason for this is the trimming effect of the shape functions, which affects specially the node chosen to do the comparison of the temperatures.

6.3.3 Cooling process

The way that each of the models capture the conduction while deforming is presented in the following summary. The figure 6.23 shows how each of the methods applied compare to the reference solution for the first 2 seconds. The relative error

Table 6.3: Results after the heat loading steps

Model	$T(t = 0.1)$	Rel. error (%)
FEM1532	834.04	
FEM110	826.78	-0.87
FEM32	769.64	-7.72
EFG110-1GP	811.51	-2.70
EFG110-3GPs	808.79	-3.03
EFG32-3GPs	781.71	-6.27
EFG32-3GPs-a3.5	702.72	-15.74
EFG32-6GPs	781.04	-6.35
RPIM110-3GPs	856.48	2.69
RPIM110-6GPs	848.10	1.69

in peak temperatures over time is represented in figure 6.24.

In the case of the FEM the heating up was well matched in the case of the 110 nodes model. After that stage, the temperature evolution shows a worse behaviour of the diffusivity process. The same effect appears in the 32 node model, although the peak temperature is smaller and cooling temperatures therefore remain below the reference solution.

Temperature evolution in the case of both meshfree methods is much more satisfactory for the 110 node model. Although starting with slightly different peak temperatures, both models converge to the reference solution, meaning that the amount of thermal energy from the loading steps has been captured properly. Also the richness of the shape functions allow to match the diffusivity behaviour with an order of magnitude less of nodes, when compared to FEM.

A special study has been made for the sensitivity of the shape function parameters for the EFG method (Figure 6.25). The model giving the best behaviour, and converging to the reference solution needs 110 nodes and 3GP per cell. If the number of points is reduced, the peak temperature is not affected (due to the linearity of the load), but the conduction process is clearly limited. When the number of nodes is reduced, there is not a satisfactory model for this problem. Several tests changing the method's parameters have been performed, showing here a small representative set. An increase of the integration points with such low number of nodes does not help in capturing with enough accuracy the temperature gradient of the diffusion process. In fact, solution drifts further apart when going from 3 to 6 GPs. If the size of the influence domain is increased, the shape functions are of course smoother, and the peak temperature error is very high. Diffusion behaviour improves but no convergence is achieved after 5 seconds, showing some oscillating behaviour instead.

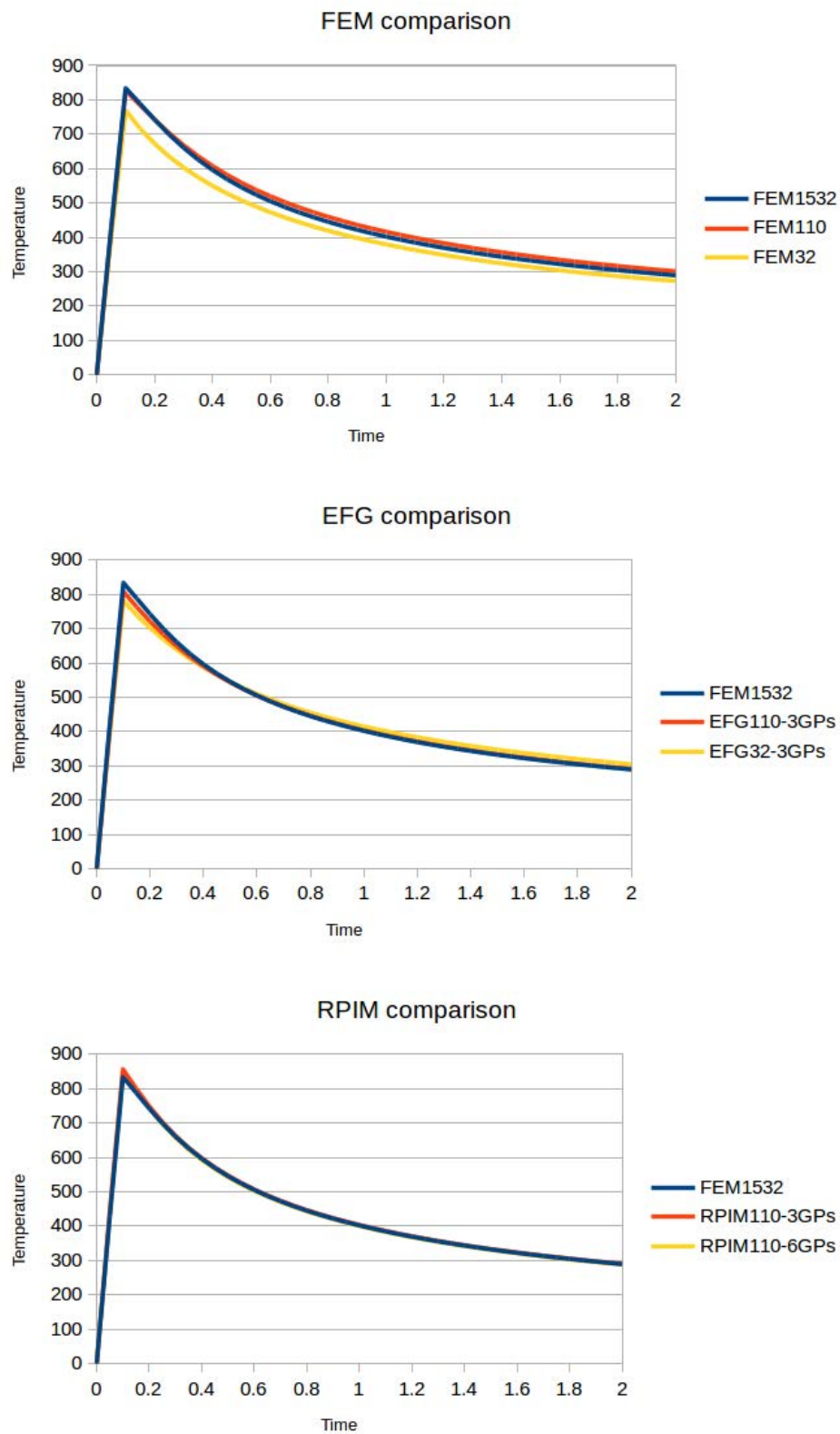


Figure 6.23: Temperature evolution compared to reference solution for the first two seconds

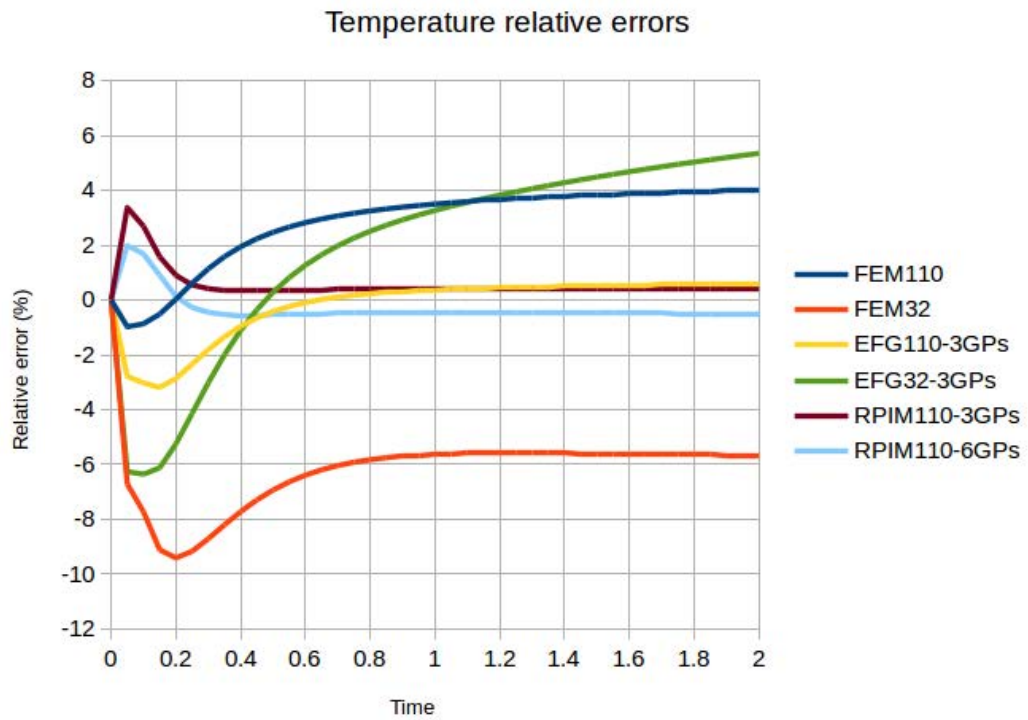


Figure 6.24: Temperature relative errors for all models

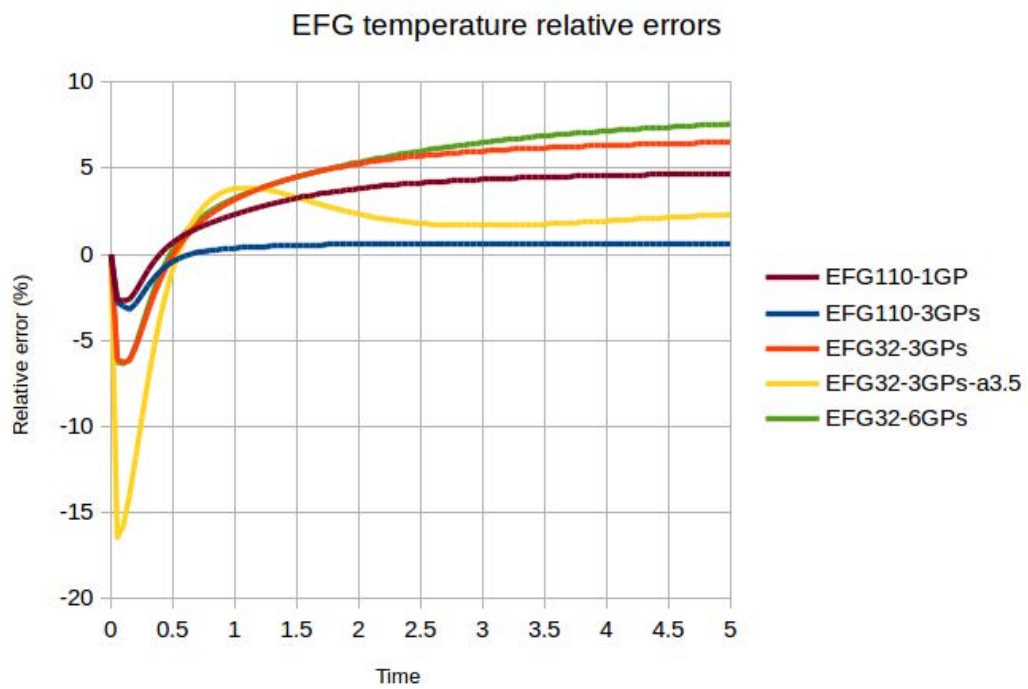


Figure 6.25: Temperature relative errors for EFG models

Application Examples and Simulation of Real Systems

7

This chapter is devoted to show potential advantages of the meshfree approach. Instead of comparing to FEM solutions, a set of examples are presented where FEM has different issues, either defining or solving the model. The second section introduces a complex research application, detailing how the meshfree methods are applied, and demonstrating how they can provide some real advantages.

7.1 Application examples

7.1.1 Generalized joints example

A very frequent issue when modelling real multibody systems is establishing the joints between their flexible components. Usually the joint defines a set of constraints in a boundary of that component rather to just a point. As opposed to rigid body parts, where the movement of any point can be extrapolated from the reference frame, which holds the degrees of freedom; flexible bodies discretized by means of FEM can only interpolate the movement inside each of the finite elements.

If the connection point of the joint is not defined inside the domain, there will no way of approximating its location using the FEM shape functions. This issue has been addressed by introducing the so called *Multi-point constraints* (MPC), which connect a set of nodes to a point inside or outside the domain. Two usual set of constraints are defined in this way:

- Displacement MPC joins the position of all of the boundary nodes to the joint definition point. This results in a rigid boundary (i.e. node-to-point distances are fixed).

- Force MPC does something similar with the internal forces instead of the displacements. This is called *flexible MPC* in some commercial implementations, as the boundary is allowed to be deformed, but on the other side the proportion of the constraint force is fixed in advance for each pair node-point; either being constant or dependant on the distance.

An alternative approach was presented in section 4.4 by using the meshfree functions for defining the joint kinematics by more than one node per body. The main idea consists in defining the position of any point using a meshfree shape function supported by a subset of nodes from the flexible domain. The joint constraints will be applied using the position of this point, which is not part of the system of equations (i.e. not a node). In consequence, the point can be anywhere, inside or outside the domain, and its location will be implicitly determined by a selected group of nodes, without any other additional constraints. From the practical point of view, the possibilities are huge, as it allows to place a joint independently on the underlying discretization of the flexible body.

An application of this generalized joint definition is presented here. A two body pendulum is studied, which is similar to the ones analysed in previous chapters. The main difference is that the flexible link has a hole representing the revolute joint that exists between this part and the rigid link. With this consideration, the internal rod will be part of the rigid link. A complex but realistic way of solving the joint's behaviour would be to model the rod and placing unilateral contact elements between the rod and the flexible body's internal boundary. Of course, this approach would increase the cost of solving the system by several orders of magnitude, and probably pose a big challenge to the solver algorithm.

The problem is tackled in this example by using MLS shape functions to approximate the position of the central point of the circular hole. The hole centre is placed in $(x, y) = (0, 0)$, and has a radius of $r = 6$. The shape function is defined using all the nodes of the flexible body within an influence domain of $r_{ID} = r \alpha_i$, with $\alpha_i = 1.1$. As the flexible domain is modelled by the mesh shown in Figure 7.1, this implies that only the nodes of the hole boundary will be selected, although no other definition nor manual input is needed.

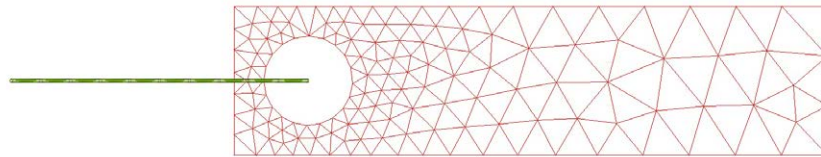


Figure 7.1: Model of rigid bar and flexible body with hollowed revolute joint

To show how flexible the approach can be, two different methods have been used for modelling the flexible domain, but neither of them is the EFG (corresponding to the MLS used for the joint). Instead, the pendulum has been analysed using FEM and RPIM methods, showing similar response with subtle differences, as it is expected after the validation and benchmarks shown in previous chapters. The evolution of the configuration is shown in Figure 7.2, where the deformation of the hole is significant. The reaction load of the joint is plotted in figures 7.3 and 7.4, showing smooth behaviour overall, with just the regular shifts in angle when the magnitude is almost zero.

This example demonstrates that the use of meshfree functions for generalized joints is perfectly compatible both with FEM and meshfree discrete solid models.

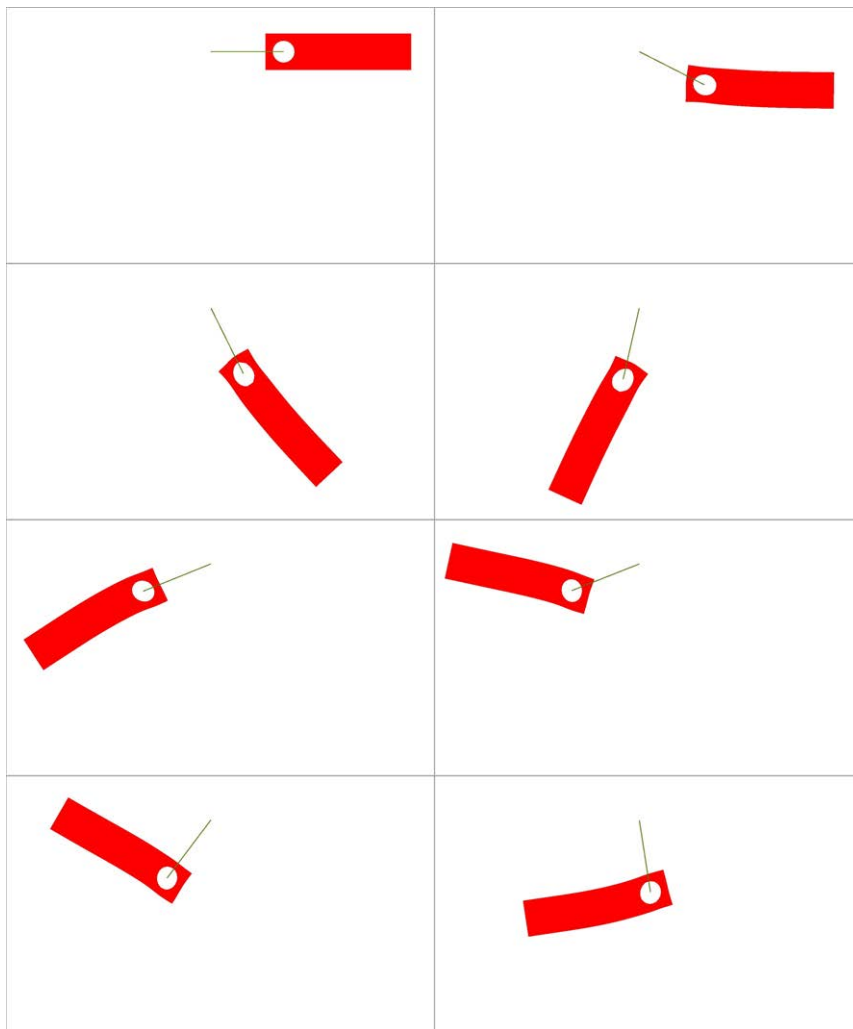


Figure 7.2: Time evolution of the reaction components for the joint between the bodies

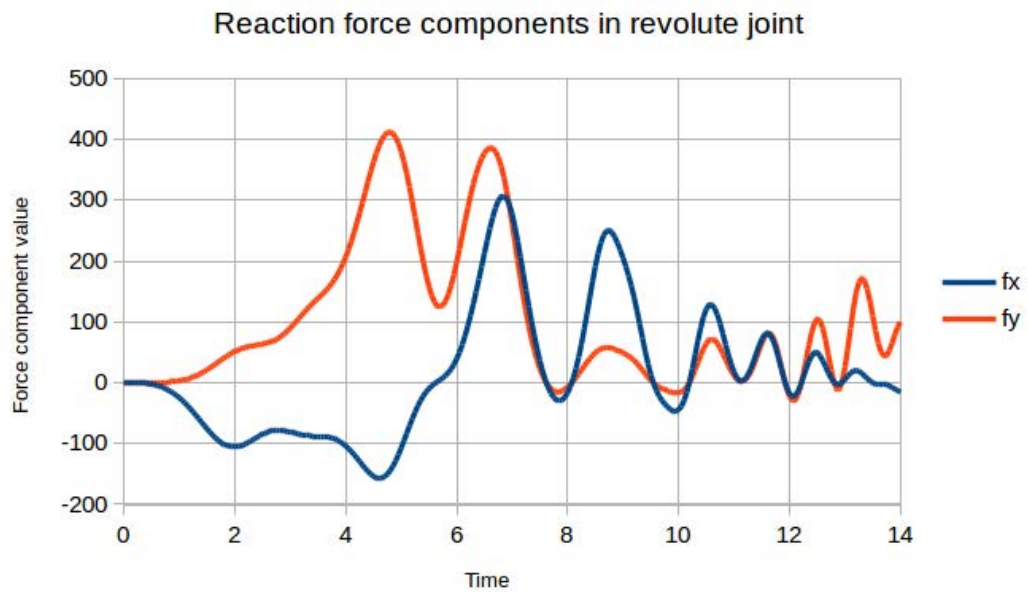


Figure 7.3: Time evolution of the reaction components for the joint between the bodies

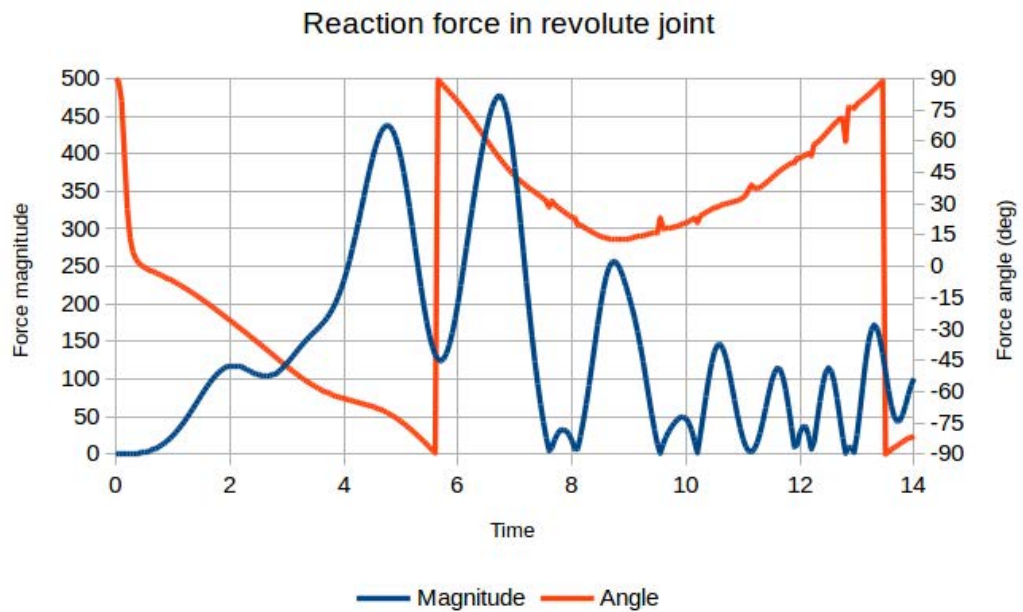


Figure 7.4: Time evolution of the reaction components for the joint between the bodies

7.1.2 Hollowed double pendulums example

In the previous chapter, a solid beam was used to model one of the links of a double pendulum under mechanical (6.1) and thermo-mechanical (6.3) effects. The 2D model of the beam used in those benchmarks was compared to a 3D prismatic beam in section 5.3.

The first section of this chapter has shown how shape functions can be used to approximate the displacement of a point even if it is located outside of the domain. This section describes a further application of this intrinsic advantage: using completely different discretizations for the support domain and the integration domain. The main objective is showing how powerful this approach can be when dealing with complex geometries, but willing to preserve a low number of degrees of freedom.

Two kind of nodal arrangements are used for modelling the hollowed beam:

- A set of scattered nodes in the convex hull of a body is used for the support of the thick wall beam model.
- Instead of spreading nodes inside the convex hull, when the void is too large (low thickness). its preferable to arrange the nodes only inside of the solid domain, as in figures 7.6 and 7.7. This approach is still useful to reduce the overall size of the system of equations, as is demonstrated with the regular and thin wall beams.

Discretizations used:

Thick model: Corresponds to a wall thickness of 3, leaving a gap between them of size 2. The nodal distribution corresponds to the same 110 nodes used in chapter 6 for benchmarks. This is a regular node distribution with average distance of 2.5 approximately. Integration cells are defined using a two-domain grid of 125 points and 158 2D cells.

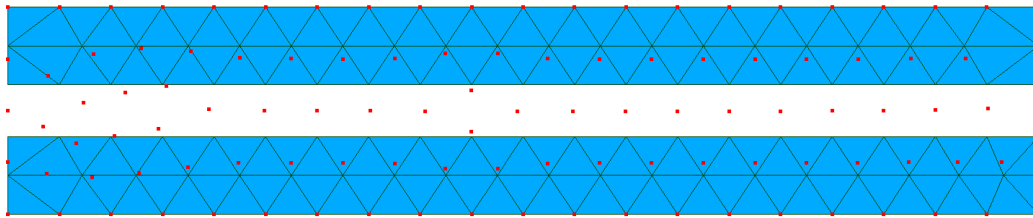


Figure 7.5: Discrete nodal distribution and integration grid for the thick hollow beam model

Regular thickness model: This is the same geometry studied in the beam validation section 5.3. The thickness of the beam was in that case of size equal to 1,

with an internal square hollow section with side dimensions of 6. The main difference between this model and the one analysed for validation purposes is that two different nodal arrangements are used for the support nodes, and for the integration cells. The shape functions are built using a reduced number of nodes, 40 in total. For the integration domain, two grids are defined totalling 116 points and 112 2D cells.

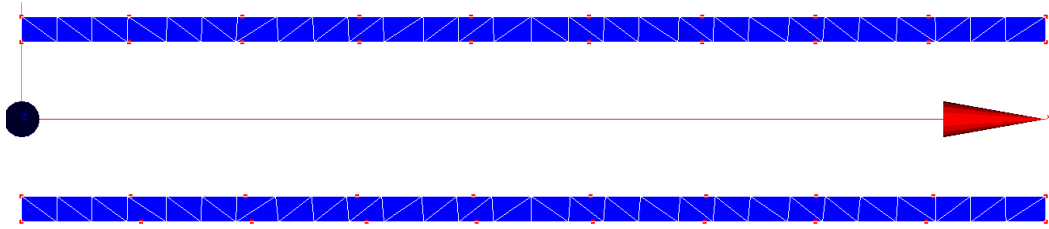


Figure 7.6: Discrete nodal distribution and integration grid for the regular thickness hollow beam model

Thin wall model: Taking the plate slenderness to the limit, a third model is proposed with a wall thickness of just 0.5, and hollow height of 7. The integration grid defines the domain with 148 points and 144 2D cells; while the degrees of freedom are concentrated in a coarse distribution of 57 nodes.

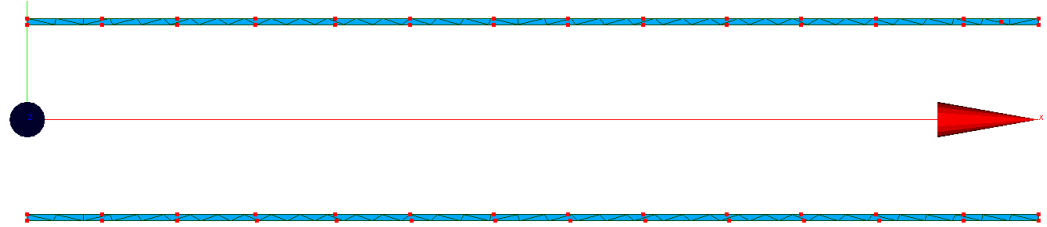


Figure 7.7: Discrete nodal distribution and integration grid for the thin hollow beam model

The three models have been used to replace the solid beam of section 6.1 using the same material and analysis parameters. The evolution of the deformation is shown in figures 7.8, 7.9, 7.10, respectively. The thinner the model, the more similar the overall beam behaviour is. This is somehow expected, as the stress in the solid domain is due mainly to bending. The deformation depends on a combination of gravity and inertial volumetric effects, and the point reaction force coming from the joint with the bar. The thinner the section gets, the more constant the strain within it is; meaning that the relation between mass and stiffness (modal content) tends to get uniform.

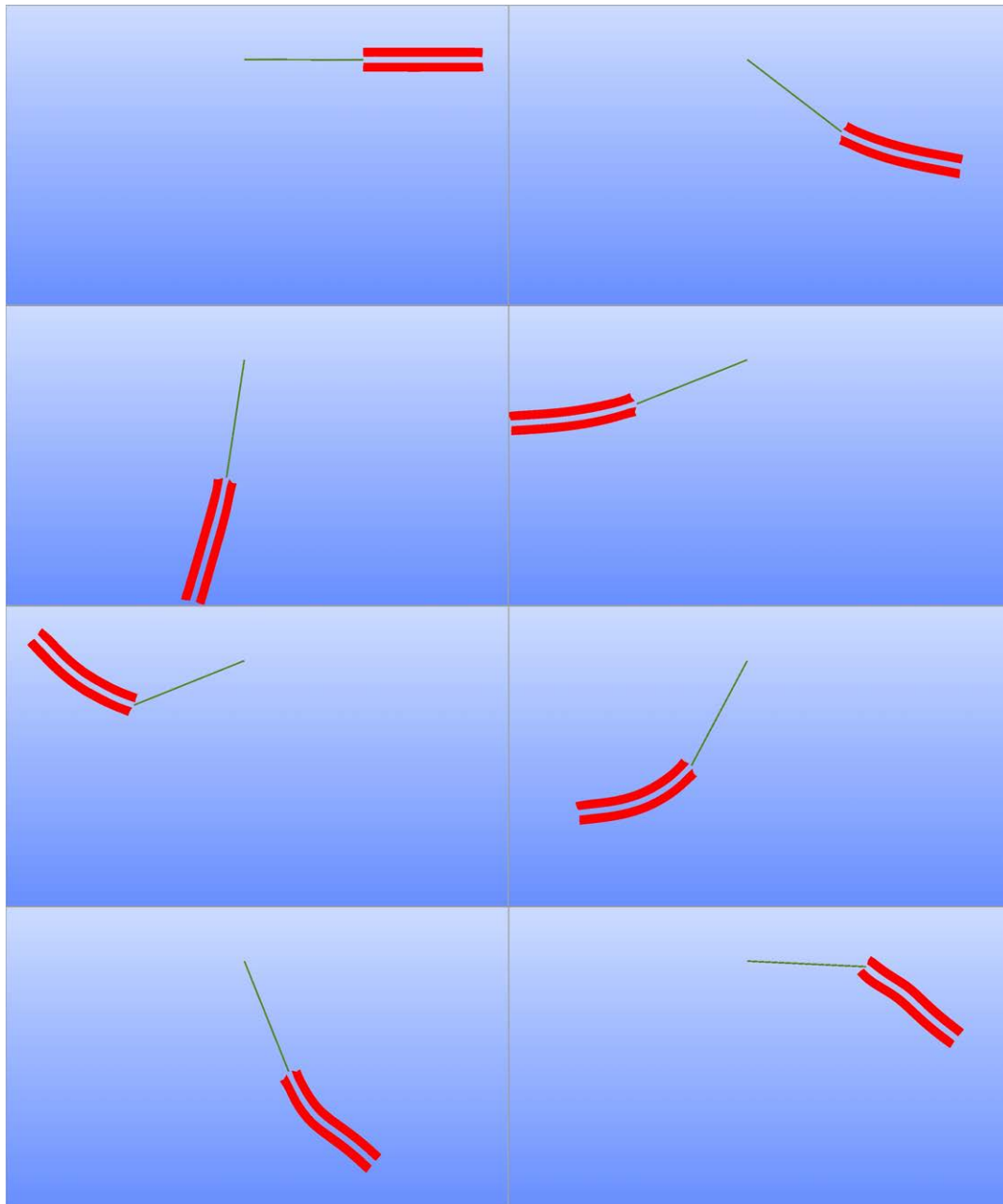


Figure 7.8: Time evolution of the configuration for the thick wall flexible double pendulum

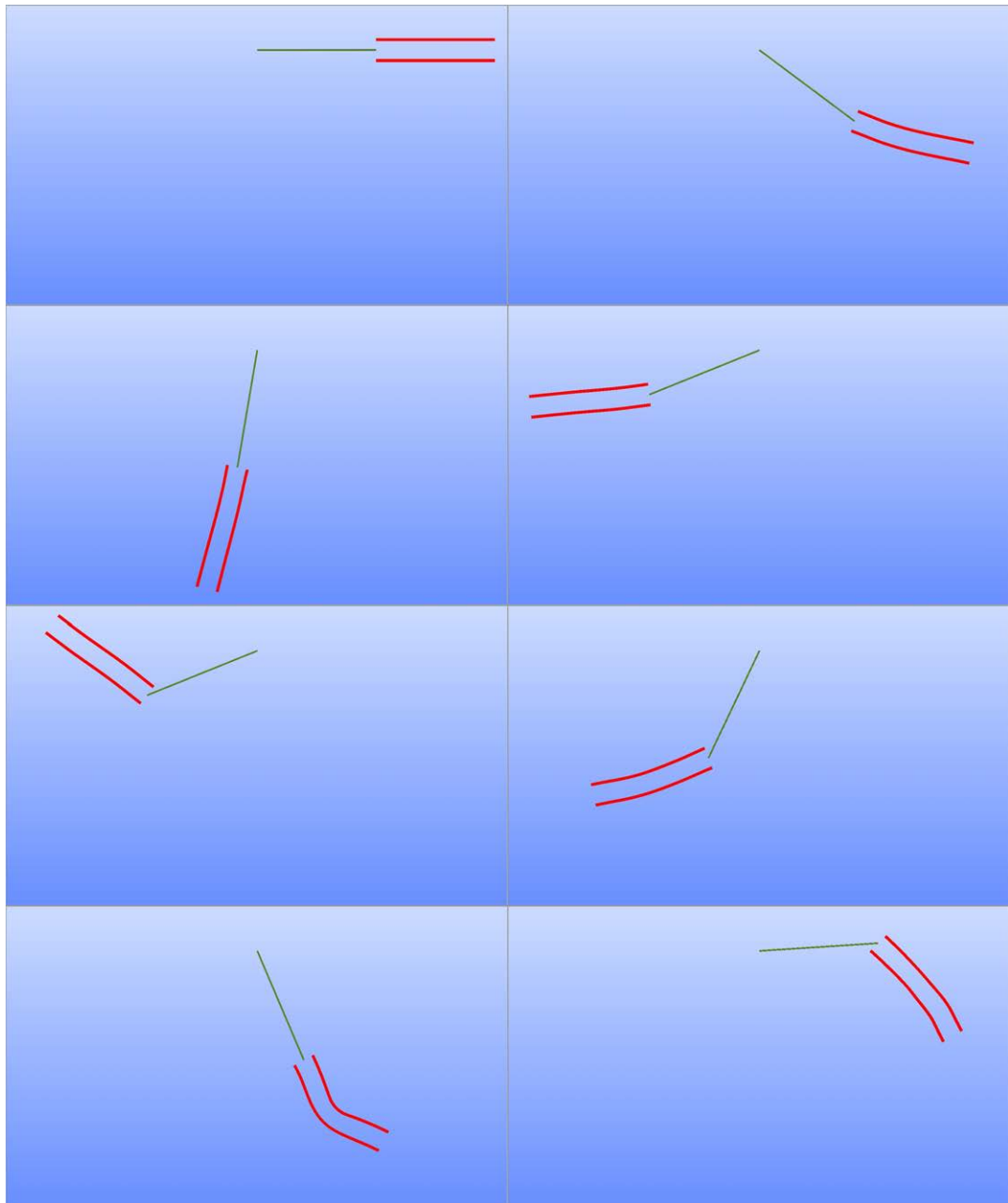


Figure 7.9: Time evolution of the configuration for the medium thickness wall flexible double pendulum

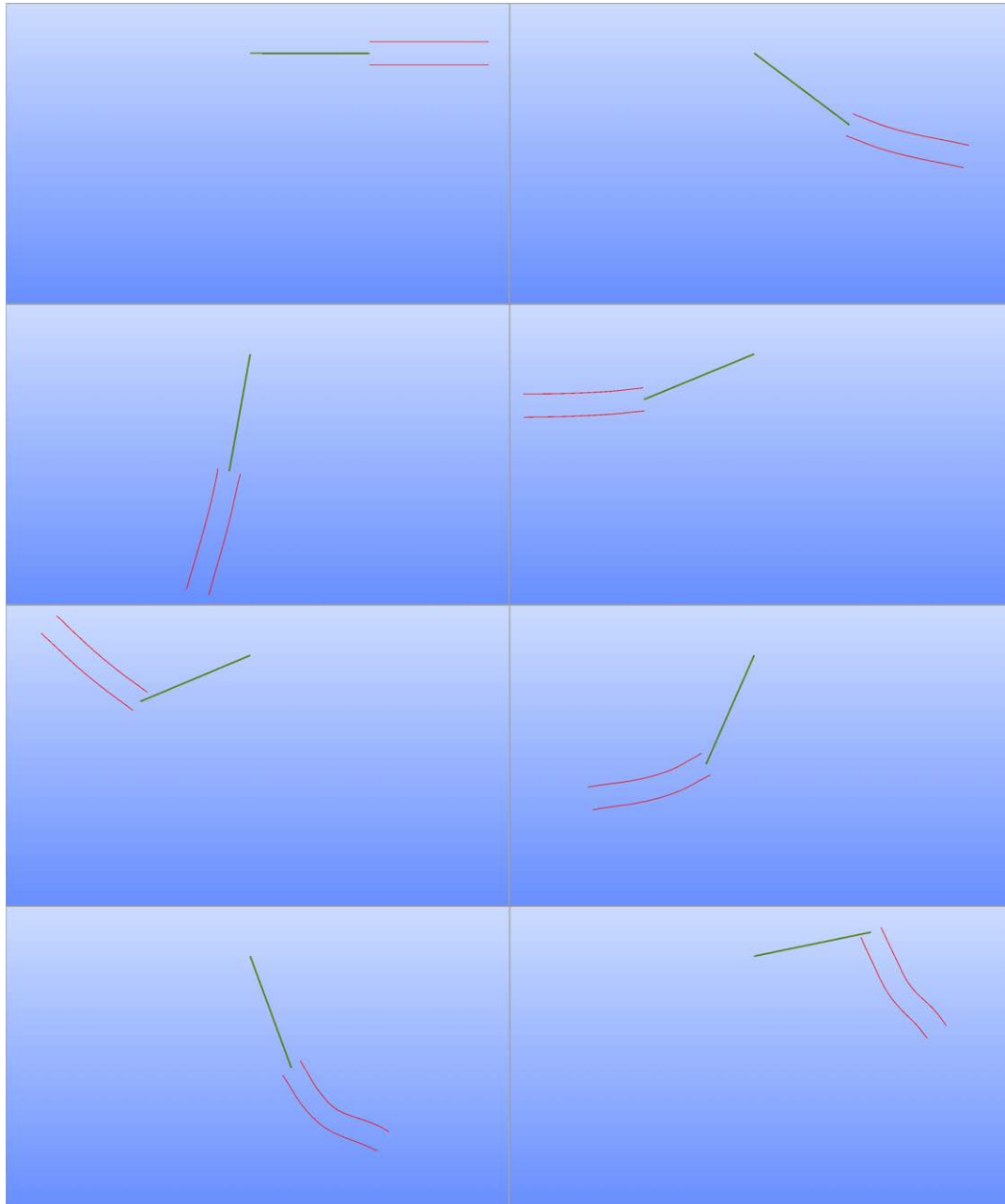


Figure 7.10: Time evolution of the configuration for the thin wall flexible double pendulum

7.1.3 Generalized contact example

A contact model that takes advantage of the meshfree node searching algorithms was presented in section 4.5. The basic problem to solve is a contact between two bodies in locations that are not known *a priori*. Two applications of are now briefly introduced, using the same material properties:

- Density, $\rho = 7850$
- Young's modulus, $E = 0.7E6$
- Poisson ratio, $\nu = 0.3$

A. Point mass contacting flexible beam

A unit mass representing a sphere falls freely due to gravity (downwards with magnitude 10), colliding with a flexible curved beam in its trajectory. The initial position of the mass centre is $(x, y) = (-40, 70)$. The thin walled beam with thickness 3, has an initial configuration of semi-arch with external radius 30. It is connected to ground by a rigid link of mass 100 and end points located at $(-20, -20) - (0, 0)$. The discrete model consists on 88 nodes and 111 2D cells/elements.

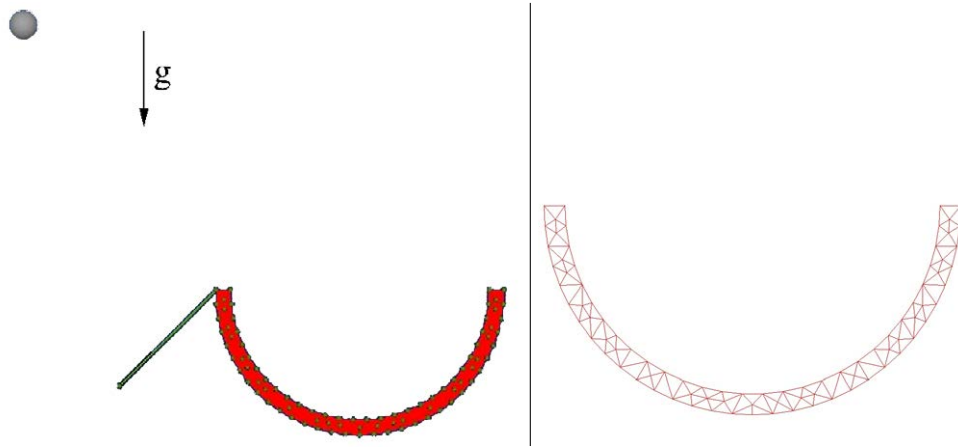


Figure 7.11: First contact example: system setup (left), and detail of the discrete domain (right)

The contact model uses a penalty coefficient of $1E7$. A triggering gap offset is defined in order to take into account the sphere radius of $r_b = 2$. This offset is only applied to the contact elements generated using the node of the point mass, although no other contacts will occur during the simulation.

Initial configuration of the system and the discretized model of the curved beam are shown in Figure 7.11. The time integration is performed with a step of $\Delta t = 0.1$, both FEM and RPIM formulations have been successfully tested. The evolution of the system is represented in Figure 7.12.

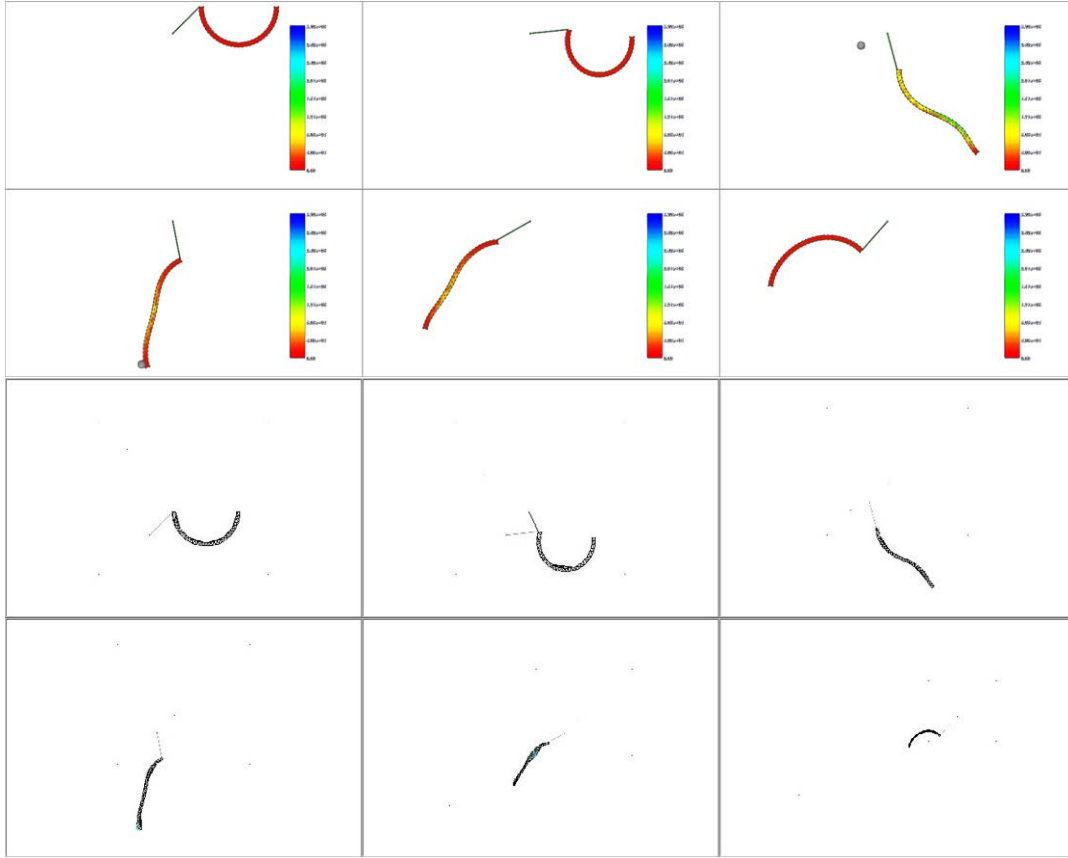


Figure 7.12: Time evolution of the configuration for the thin wall flexible double pendulum: From top to bottom, left to right, snapshots for six intermediate configurations showing Von Mises stress; and same snapshots representing the automatically generated contact elements

B. Rigid bar contacting flexible solid

A rigid bar of mass 100 is dropped inside a flexible container which wobbles due to inertial effects. The solid is fixed to the ground in its four exterior corners, and a thickness of 5. The same value as the thickness is chosen for the contact α_c parameter.

As before, a penalty contact model is applied, with a factor of 10E6. A lower value of the penalty along with a higher mass (and kinetic energy) of the bar, will facilitate observing the penetration of the nodes in the solid. Higher factors have been also tested, requiring a reduction in the time stepping (Euler implicit integrator) in order to avoid artificial numerical impulse. A variable step size integrator with predictor/corrector scheme would have allowed increasing the automation and speed of the simulation procedure.

The simulation is schematically shown in Figure 7.13. The bar is at an initial horizontal distance to the internal edge of the solid of 4, so contact elements to this side are automatically created and destroyed during the free fall of the bar. It has

been checked that none of them influences the movement of the bar, as their height is over the contact gap tolerance. Also, contact elements are automatically created in the internal corners of the flexible bodies, suggesting that self-contact constraints would be applied in case it suffered very large deformation.

As the bar approaches the inner horizontal boundary of the solid, a set of contact elements are created, activating the constraint force upon collapse. The gap remains negative for more than one step, warranting that the simulation does not blow-up, but not enough to restore the initial energy to the bar, which raises less than half way up in the air again.

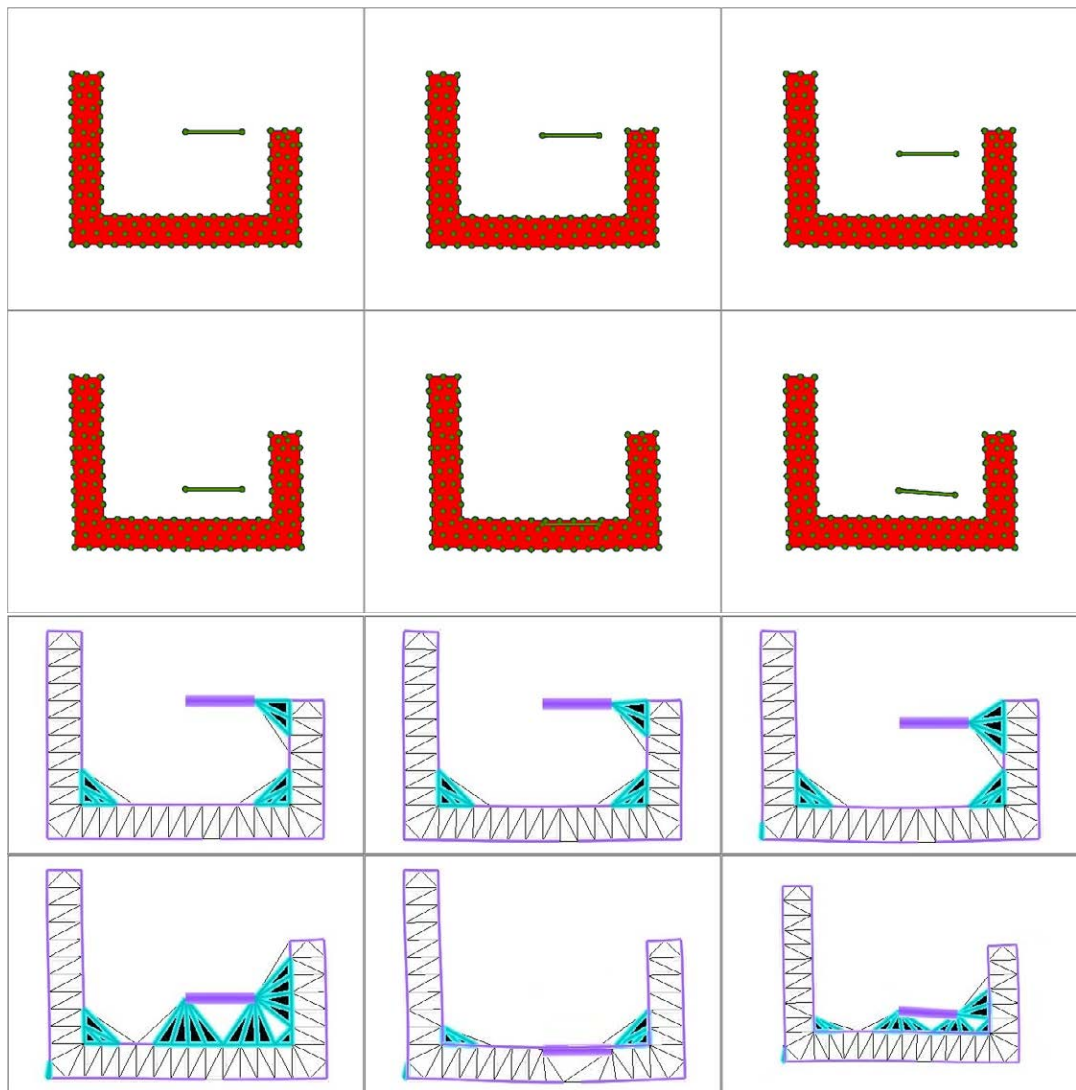


Figure 7.13: Time evolution of the configuration for the thin wall flexible double pendulum: From top to bottom, left to right, snapshots for six intermediate configurations; and same snapshots representing the automatically generated contact elements

In summary, two simple applications have been presented that show automated

contact using the same node search algorithms employ to build the meshfree shape functions. The formulation of the contact itself is quite simple, but robust if the integration parameters are selected taking into account the relation between maximum kinetic energy and contact penalty coefficient.

7.2 DEMO Remote Handling for blanket segments replacement

The future DEMO (fusion demonstrator) tokamak is in its conceptual design stage, which will conclude at the end of this decade. The baseline design is composed of a set of sectors (between 16 and 20) which repeat all around the machine forming the donut shaped tokamak. The in-vessel components need regular replacement due to accumulated neutron damage, and are accessed through three ports located in upper, equatorial, and divertor (bottom) positions for each port.

7.2.1 System description

The most massive in-vessel components are the blanket segments, also called Multi-Module Segments (MMS in Figure 7.14). These structures can be up to 10 meters tall, and weigh between 50 and 80 tonnes. For each sector, there are three outboard segments: central (COBS), right (ROBS), and left (LOBS); and two smaller inboard segments: right (RIBS), and left (LIBS).

The main challenge of the replacement of the MMS is related to the integration of the machine requirements. Basically, the blankets need to be as close together as possible, and the ports be as compact as possible. These translate into very tight clearances (20mm is the present goal), which leads to complex transport kinematics, and also extremely precise heavy-duty remote handling equipment.

This section is devoted to show how the previously presented methodology can benefit the design and assessment of the remote replacement of the blanket segments.

The basic design of the blanket segments is composed on a back-structure (BS) and a set of (6-10) blanket modules attached to it. All the service connections run from the ports' infrastructure (pipes and cables) all the way inside the BS to each one of the modules. Those modules are not mechanically interconnected between each other, leading to a more flexible structure than would be thought. It is worth noticing that due to extremely harsh radiation inside the vessel, the blankets need to be handled right from the top and bottom, avoiding direct exposure of the mechatronic equipment to the core of the gamma radiation field.

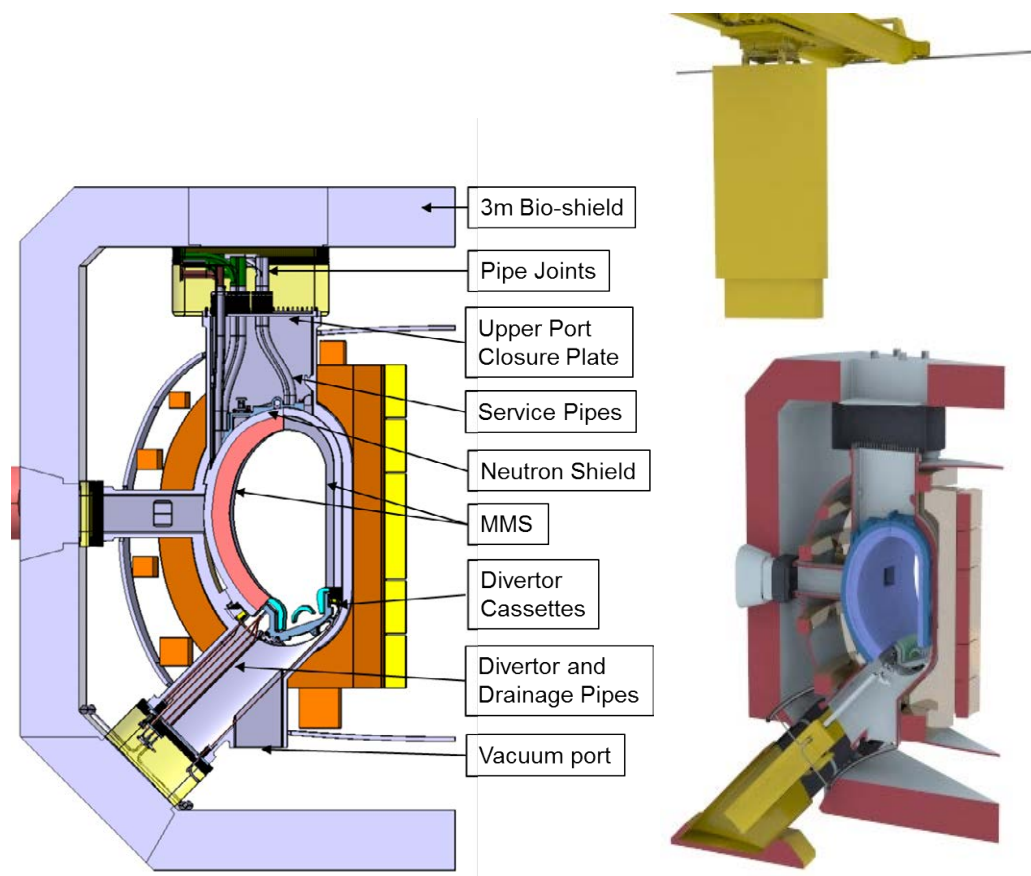


Figure 7.14: DEMO cross section (left) and cask deployment in ports (right)

7.2.2 Preliminary assessments and problem statement

During the concept design stage, several basic kinematic approaches are evaluated using CAD models, as shown in Figure 7.15. After important engineering details are added to the model, the movements are defined, and the complete geometry and kinematics are evaluated in a Virtual Reality mock-up, as shown in Figure 7.16.

After checking the estimation of the clearances in VR, a set of static structural analysis were performed for the blanket segments. Those confirmed the expected deformation as number one risk to induce clashes and seizures, which would have a really serious impact in the operational availability of the whole power production plant.

A two stage approach has been followed in order to evaluate the nonlinear dynamic effects of when handling the blankets:

- 2D simulation of the COBS, validation to FEM results, and comparison with rigid body behaviour.
- 3D simulation for COBS, comparing the modelling approach with the 2D re-

sults, and comparison with rigid body behaviour.

The objective is to create simulation models, as efficient as possible, aiming to the implementation of an enhanced real-time VR structural simulator.

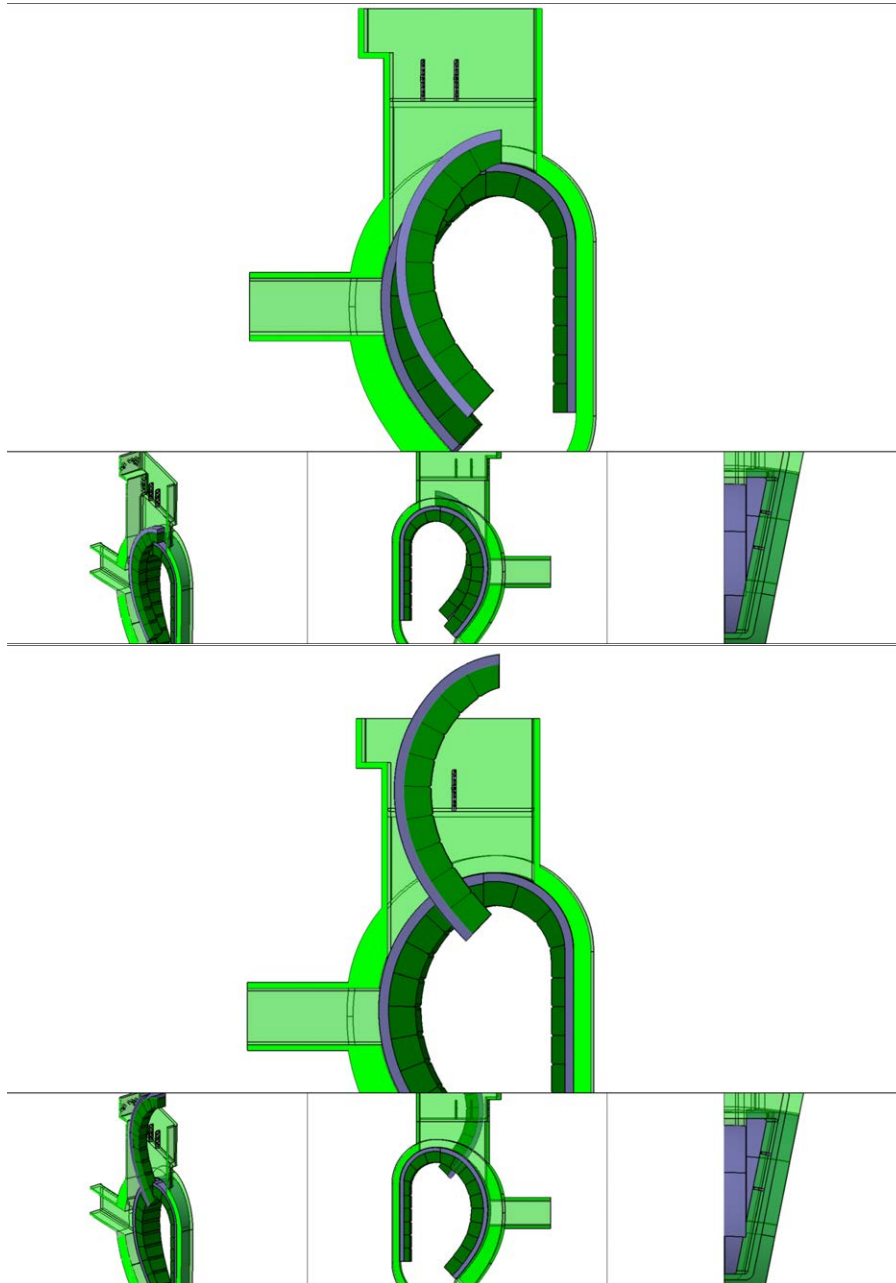


Figure 7.15: DEMO cross section basic strategy

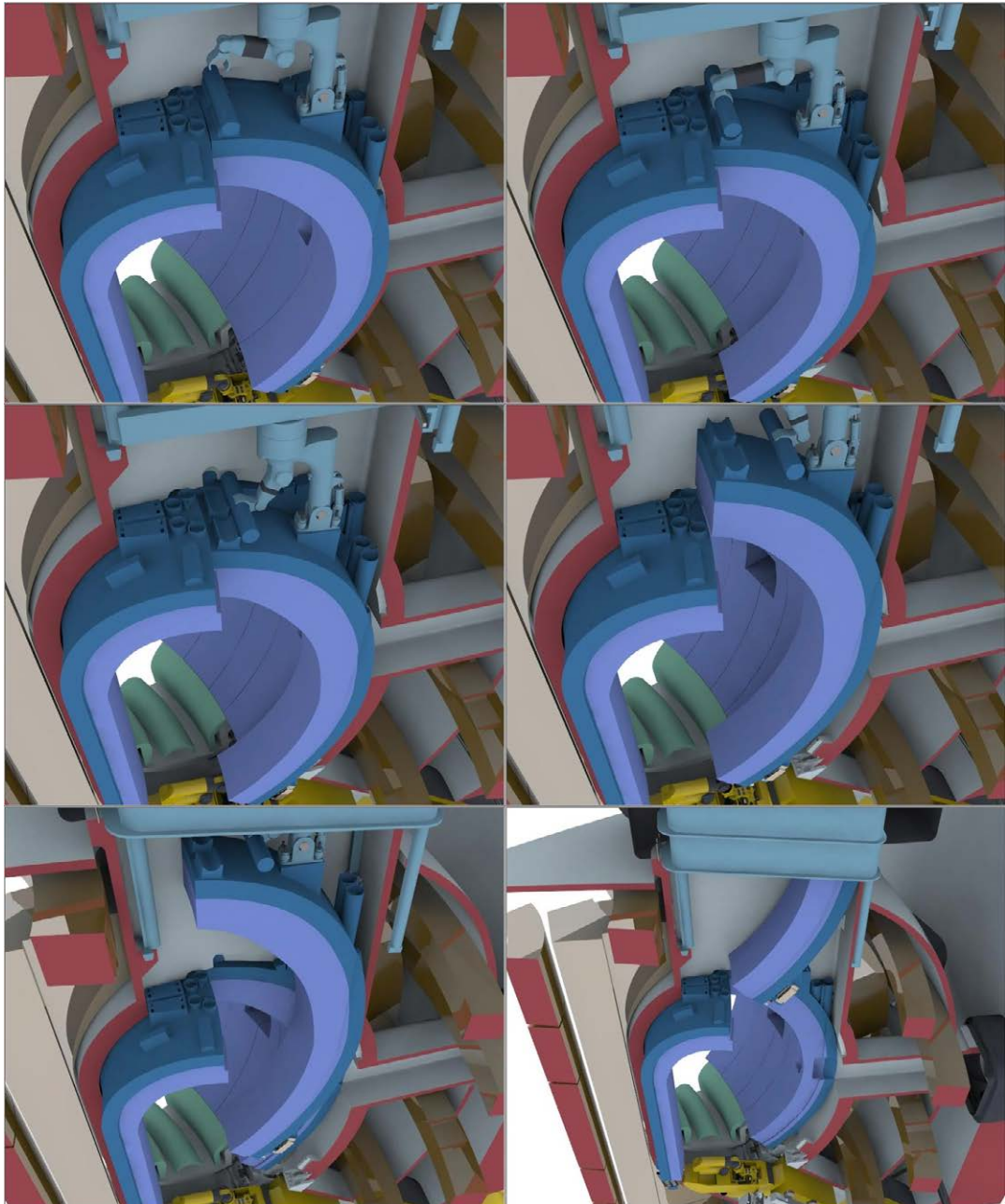


Figure 7.16: Virtual reality kinematic study for the DEMO segments handling

7.2.3 Discrete models

A simplified version of the CAD models is used in terms of design details. The maximum number of 10 blanket modules per segment is chosen, which will pose a more difficult problem. The interface and BS of the blanket segment is kept as a solid body, until more detail of the design is available. Of course, these structural members will not be completely solid due to services and weight constraints, so the material properties need to reflect this uncertainty.

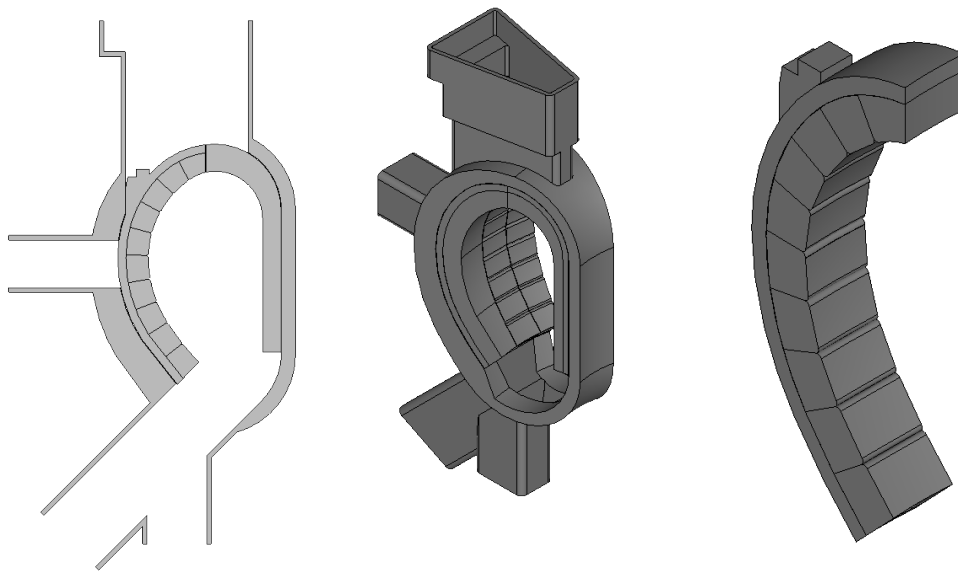


Figure 7.17: CAD models for DEMO blanket segments

The models used for each of the simulations are all composed of non-conforming meshes. As it's often the case when the CAD geometries are complex, building a continuous mesh for all the domain is something that requires some processing of the CAD shape, or—as in this case—a complete re-drafting can be necessary. Figure 7.18 shows how the different parts of the models do not match with each other, and even the lower two blanket segments are completely overlapped. For a regular one part FEM analysis, quite a lot of effort would be necessary even to have an initial rough response of the system.

The modelling is approached without any processing of the initial geometry, requiring individual discrete models of each part, which need to be joined together. The following four discrete systems have been generated:

2D plane strain models (Figure 7.18):

- Coarse Meshfree simulation model, with only 161 nodes, 157 2D cells.

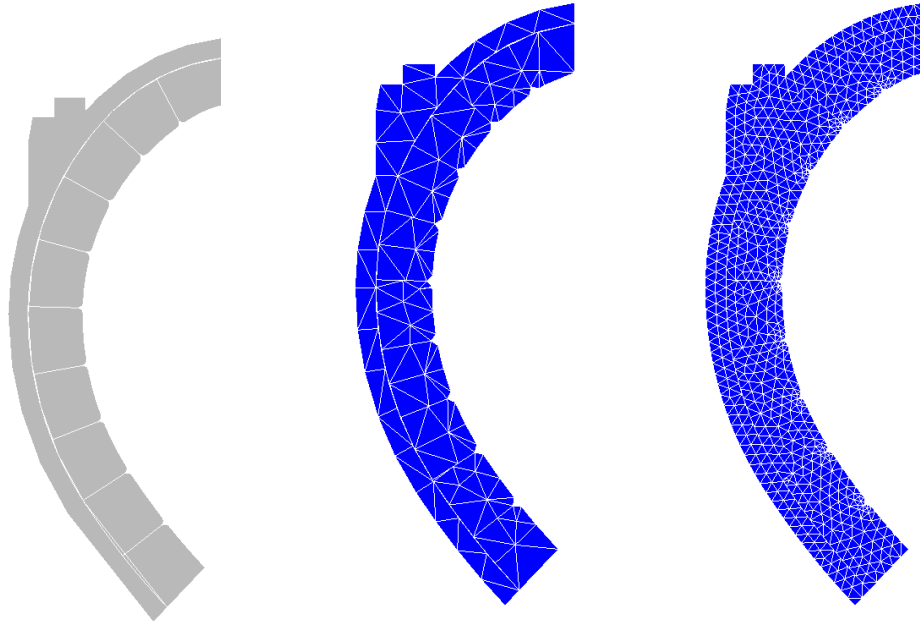


Figure 7.18: 2D section: CAD and discrete models

- Dense model for FEM analysis and comparison, with 809 nodes, and 1184 2D elements.

3D models (Figure 7.19):

- Coarse Meshfree model with 226 nodes, and 439 3D tetrahedrons.
- Denser model for convergence studies, with 497 nodes, and 941 3D tetrahedrons

7.2.4 COBS removal analyses

The 2D and 3D simulations are detailed in this section. When possible, comparison have been made with FEM discretization. The aim is to demonstrate the applicability of the meshfree methods to the conceptual study, so material properties have been chosen intentionally to produce very flexible bodies. Probably, as the deformations obtained are too large, the final design will be stiffer. Still, the conclusions about the advantages of the presented approach will still hold.

Therefore, material properties definition are based on the expected average density and worst case stiffness. Its also been considered part of the body mass not being structural, setting the following material values for all bodies:

- Density: $\rho = 3240.36 \text{ kg/m}^3$.
- Young's modulus: $E = 0.7\text{E}10 \text{ Pa}$.

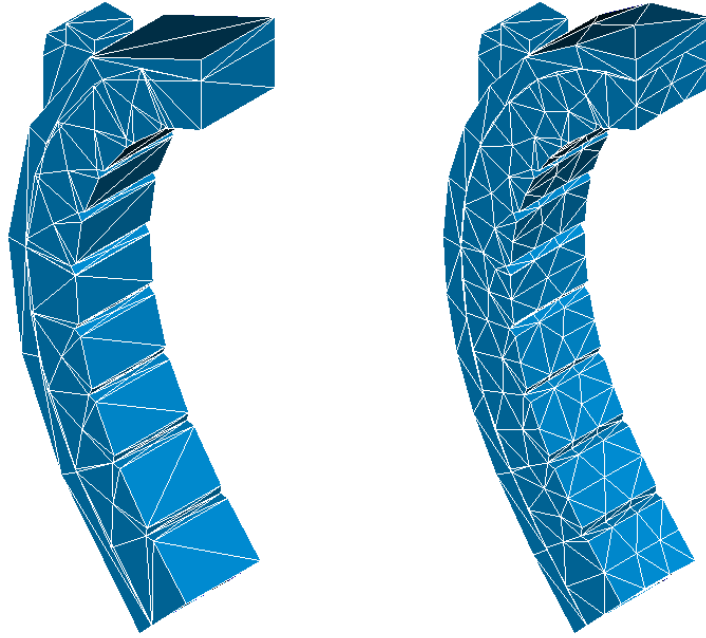


Figure 7.19: 3D discrete models

- Poisson coefficient: $\nu = 0.3$

In all cases, the remote handling interface of the blanket segment is defined in the two (for 2D) or four (3D) points of the upper-external grab feature, as shown in Figure 7.20. The motions defined for each interface point (or pair of points in 3D) is interpolated from Table 7.1, using subscript 1 for the outboard node, and 2 for the inboard. In the 2D simulations, the z values of the table applied to the y -axis.

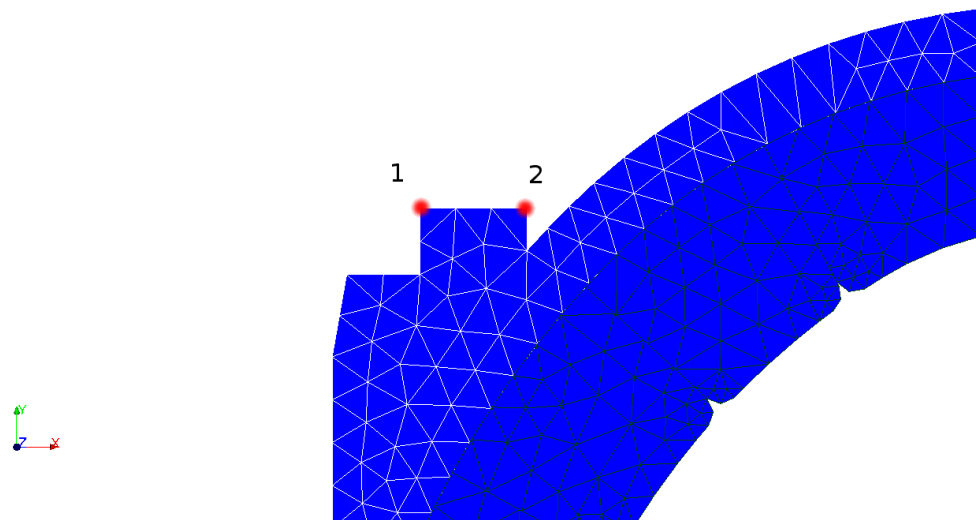
A. 2D models

Two different joints are applied to the interface points: a revolute joint for the inboard, and a sliding joint (along x -axis) at the outboard point. The mesfree model uses EFG shape functions and all the integration cells for the blanket modules have been defined with 1 GP, which speeds up computation. In this case EFG are preferred over RPIM as it behaves better when the number of supporting nodes is low. For the BS, as strain levels are larger, 3GPs are required in order to obtain enough accuracy. The values for domain of influence have been set to $\alpha_i = 2.0$ for the BS, and $\alpha_i = 2.5$ for all the blanket modules.

In the case of the FEM model, the modules are connected to the BS using double-point fixed-distance constraints. These are applied between the corners of each

Table 7.1: Imposed motion at the Remote Handling (RH) interface points

Time	x_1	y_1	z_1	x_2	y_2	z_2
0	0	0	0.0	0	0	0
15	0	0	0	0	0	0
20	0	0	0.0	0	0	0.05
45	0	0	1.5	0	0	1.55
85	1	0	4.0	1.0	0	4.05
95	1	0	4.0	1.0	0	3.95
160	1	0	10.0	1.0	0	9.95

**Figure 7.20: RH interface points defined in all models**

module and the closest node of the BS (Figure 7.21).

The analysis is performed using a BDF-2 integration method and fixed 0.25 s time step, without any contacts being considered. An overview of the motion is described with snapshots in Figure 7.22. In there, both the ideal rigid body motion and the deformed system is overlapped.

Two critical points have been identified:

- The back structure clashes with the vacuum vessel at the start of the lifting operation. This is due to the deformation reducing the effective rotation angle given to the MMS before it's lifted.
- On the other side, deflections avoid a clash between the COBS being extracted and the inboard segments, as shown in the last capture of the handling se-

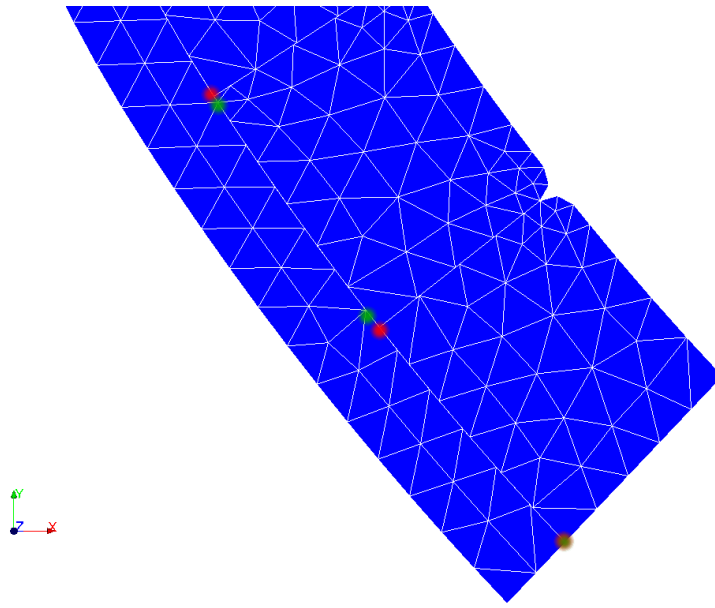


Figure 7.21: Double-point fixed joints for FEM model: modules's nodes in red, and back-structure (BS) nodes in green.

quence.

The loads that would be applied to the remote handling end-effector providing this motion are influenced by the deformation too. Figure 7.23 shows the horizontal and vertical components of those loads. Both oscillations are due to the oscillation of the position of the Centre of Gravity, and peak at every change of direction of the motion.

Total vertical reaction is equal to the total weight of the MMS. A few ripples indicate some (low) numerical errors. Being a plain strain model, all the loads need to be scaled by the thickness of the COBS (1.3 m) in order to compare them to the 3D simulations, and therefore to the actual MMS, which weighs 67 t.

The deformation of the bottom side of the MMS is compared between the FEM (Figure 7.25) and Meshfree (Figure 7.24) models. The former shows a more stiff behaviour than the later, with respective peaks of 13 and 22 cm, neglecting the initial oscillation. The vibration at each change of motion path is important, peaking 3 and 4 cm for each model respectively.

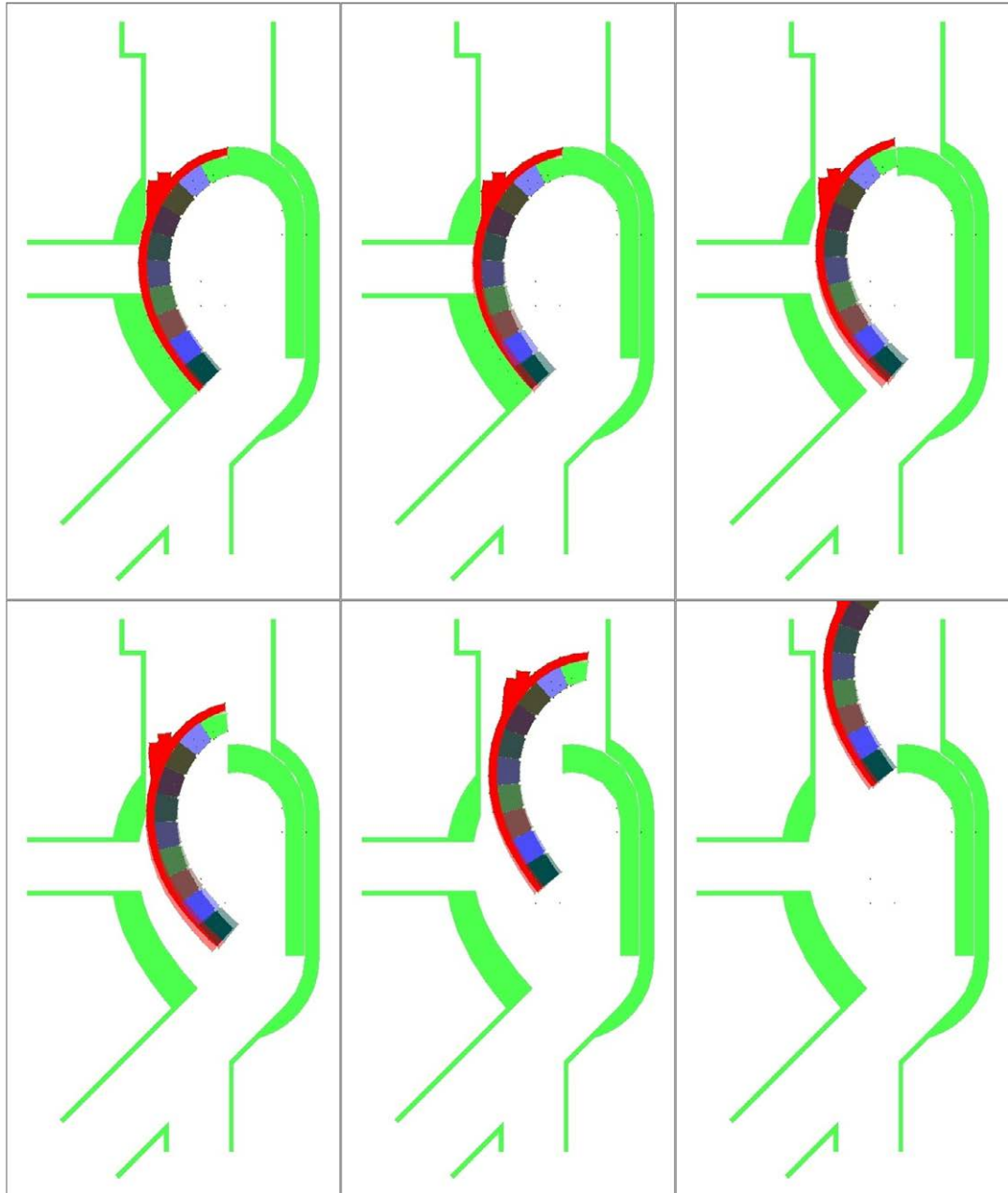


Figure 7.22: 2D comparison (superimposed images) between solid and flexible dynamics for the DEMO segments handling. Top to bottom, left to right: configuration of the COBS blanket segment during extraction operation

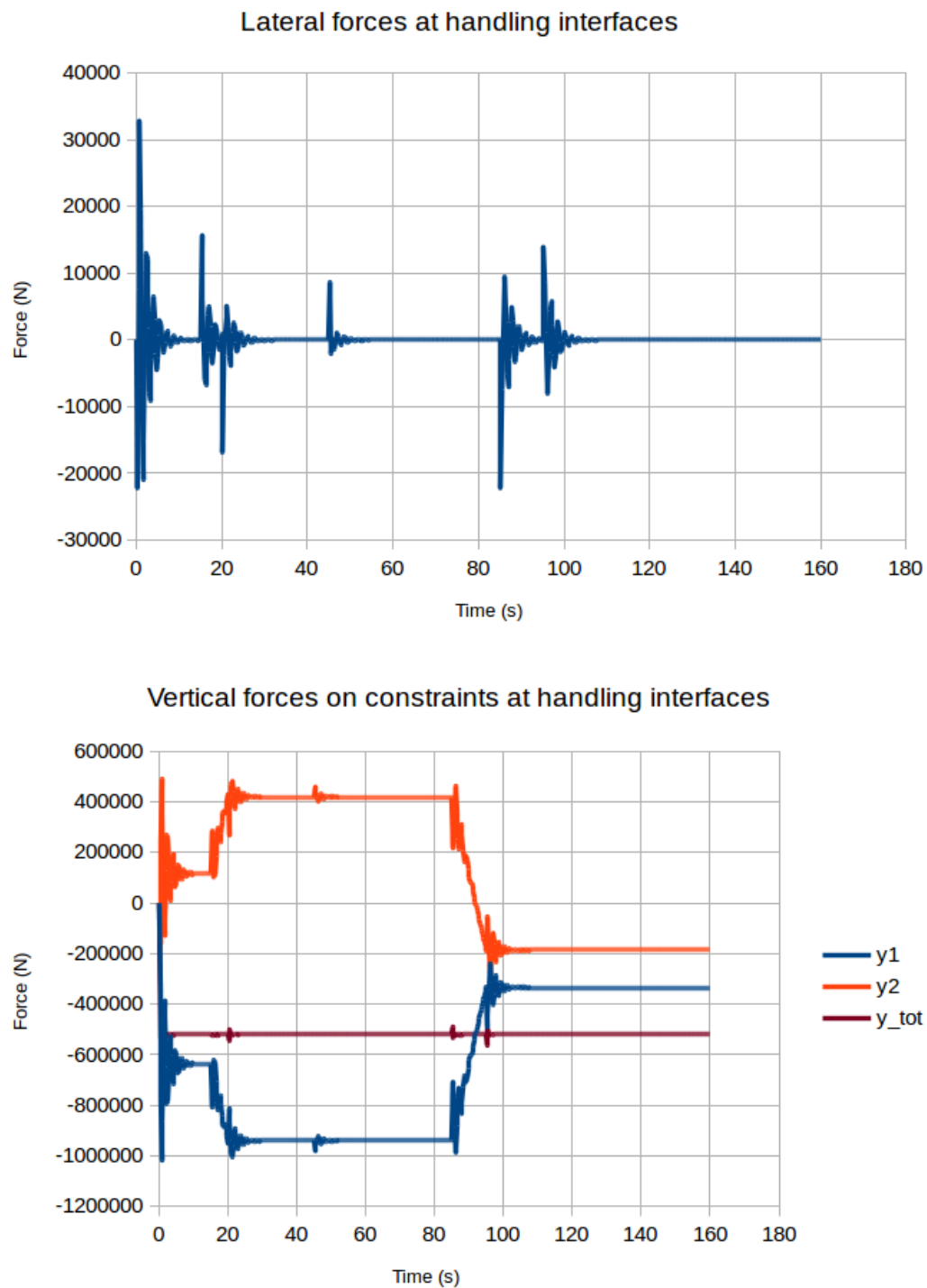


Figure 7.23: Forces for the 2D model RH interface during the extraction of the COBS

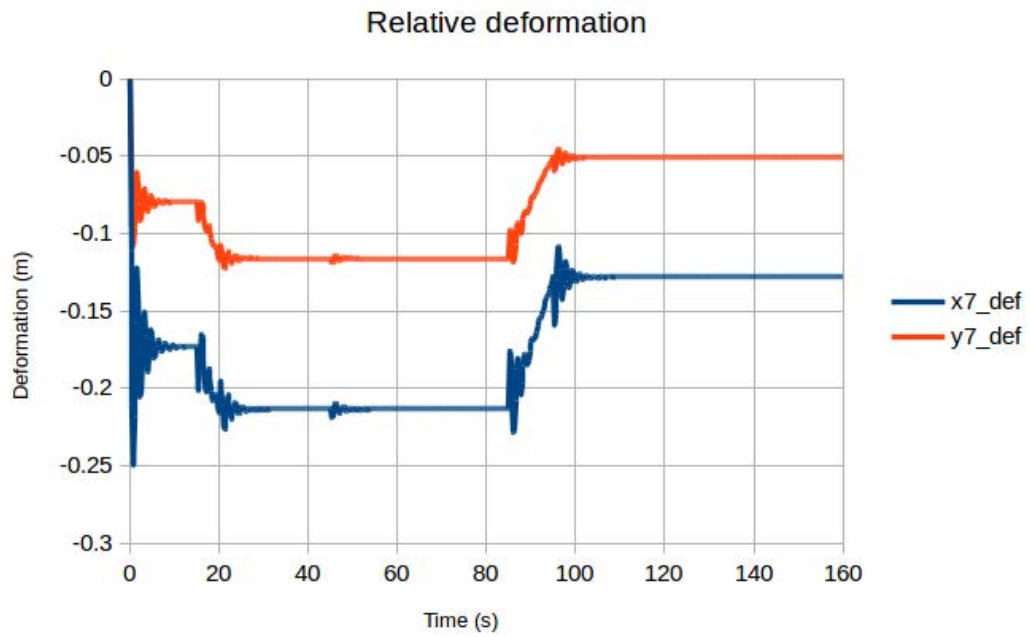


Figure 7.24: Deformation of the bottom of the MMS for EFG 157 nodes model

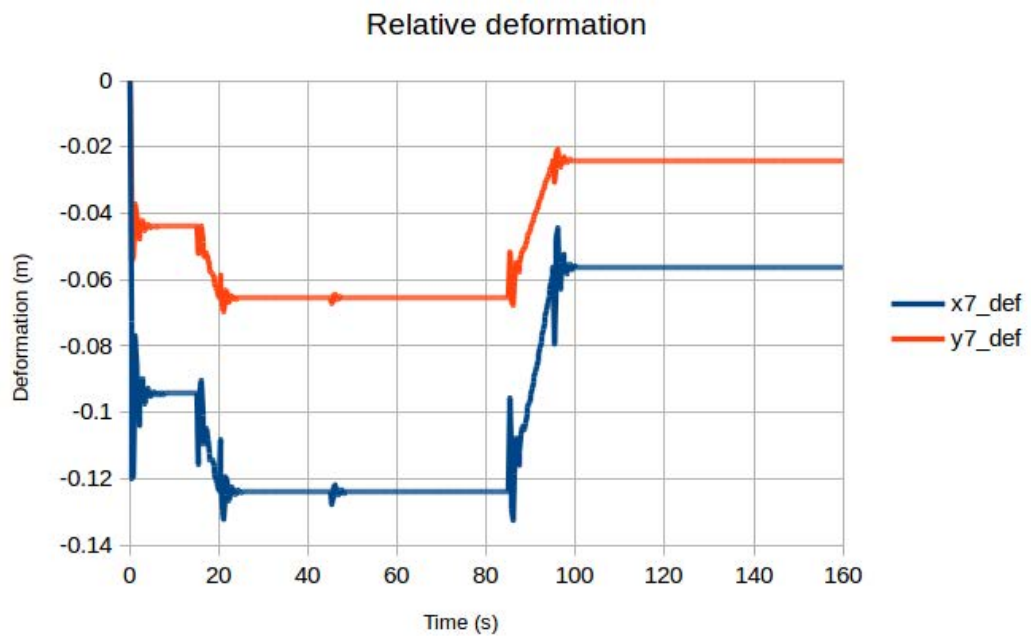


Figure 7.25: Deformation of the bottom of the MMS for FEM 1184 nodes model

B. 3D models

The 3D models are more involved to prepare, as more joints need to be defined to the Remote Handling (RH) interface. Vertical displacements need to be imposed to each of the corners of the RH interface, but also the lateral movements and rotations shall be controlled. To avoid over constraining the system, only one spherical joint has been applied to a node (inboard-left). In addition, one sliding joint in vertical direction is applied to the opposed node (outboard-right), constraining $y - z$ displacement. Finally, two axial constraints only acting in z -axis direction are also applied to the other two nodes.

The EFG parameters used in this case are a constant value for the influence domain of $\alpha_i = 2.5$, and just 1 integration point (GP) for all cells. Several runs have been performed in order to set these values; one of them using 4 GPs per cell showing no improvement in the accuracy of the model.

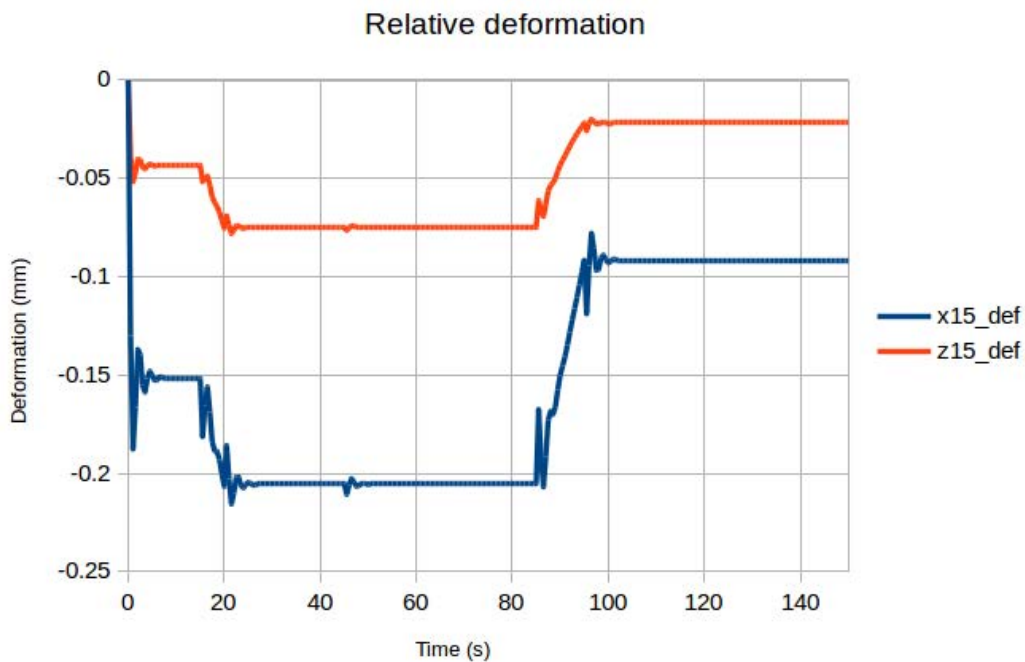


Figure 7.26: Deformation of the bottom of the MMS during the extraction of the COBS

The relative deformation of the lower node of the segment is shown in Figure 7.26. The response gives very similar results in the horizontal deformation to the ones presented for the 2D meshfree plane strain model. Differences in the vertical deformation are significant, and probably due to the lack of refinement of the 3D model in the RH interface area.

The evolution of the model is shown in Figure 7.27. External cables have been added to the representation, although their flexibility is not considered in order to

be able to compare to the previous 2D analysis. Of course, when used, as they only work in tension, a different set of joints would need to be provided in order to absorb the torque (7.29) due to the offset of the MMS CoG(7.30).

The lateral and vertical forces which the RH end-effector would need to resist are plotted in Figure 7.28. The lateral forces are mainly due to the lack of symmetry of the model and also the boundary conditions, but still remain in the order of what was obtained for the 2D model.

The vertical forces correspond to compression in the outboard side and tension in the inboard (closer to the CoG). The later become compression during the final lift, which can be an important fail-safe feature to take into account for the design of the end-effector. Its worth noting that the loads applied to each pair of nodes with equal motion are not exactly the same, specially when the CoG's offset is maximum, also due to the lack of symmetry; but overall the results are consistent and no particular flaw has been produced by such a coarse discrete model.

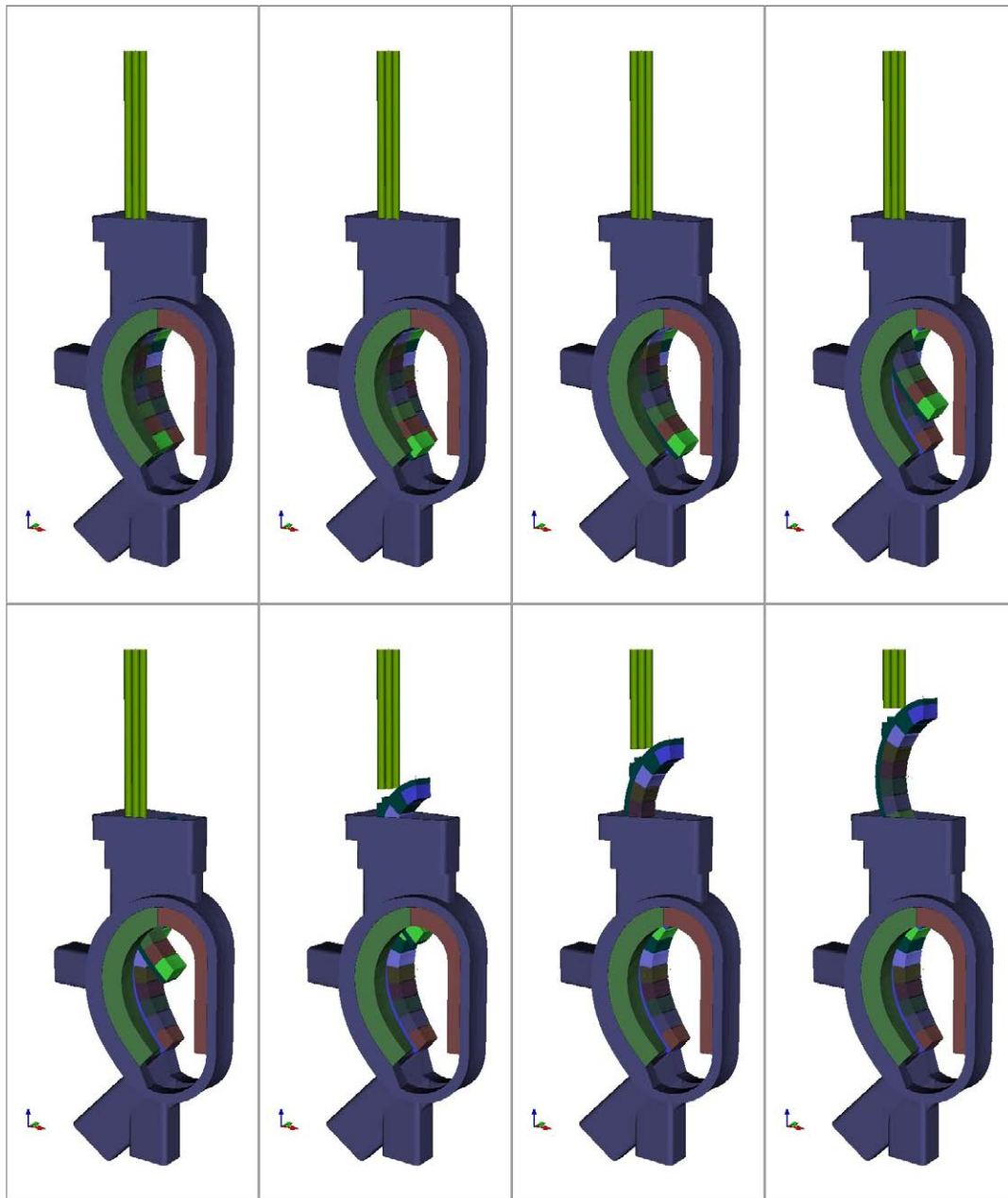


Figure 7.27: 3D multibody simulation of the flexible COBS MMS, with rigid vacuum vessel and other MMS in place. Top to bottom, left to right: configuration of the blanket segments during extraction

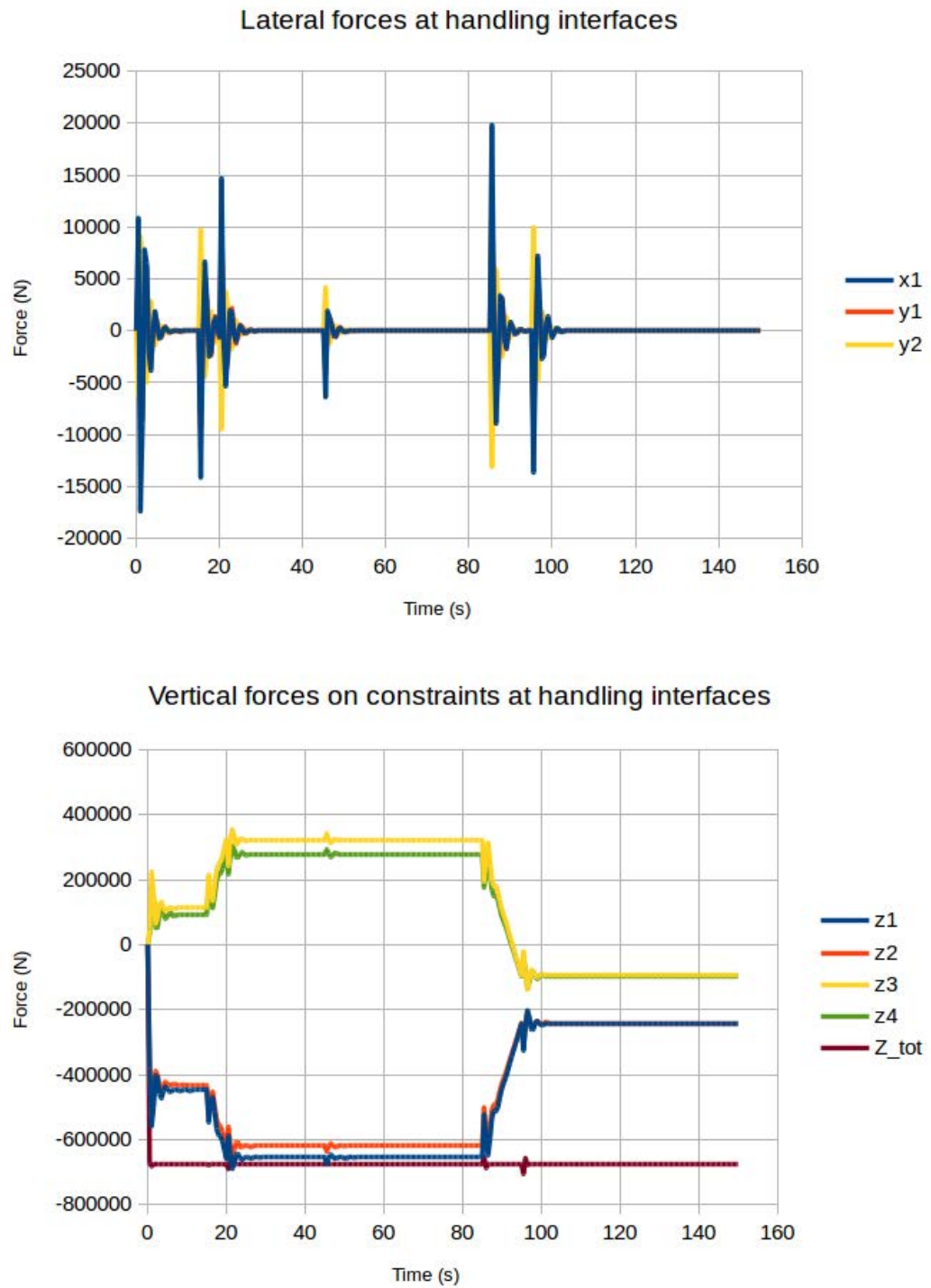


Figure 7.28: Forces for the 3D model RH interface during the extraction of the COBS

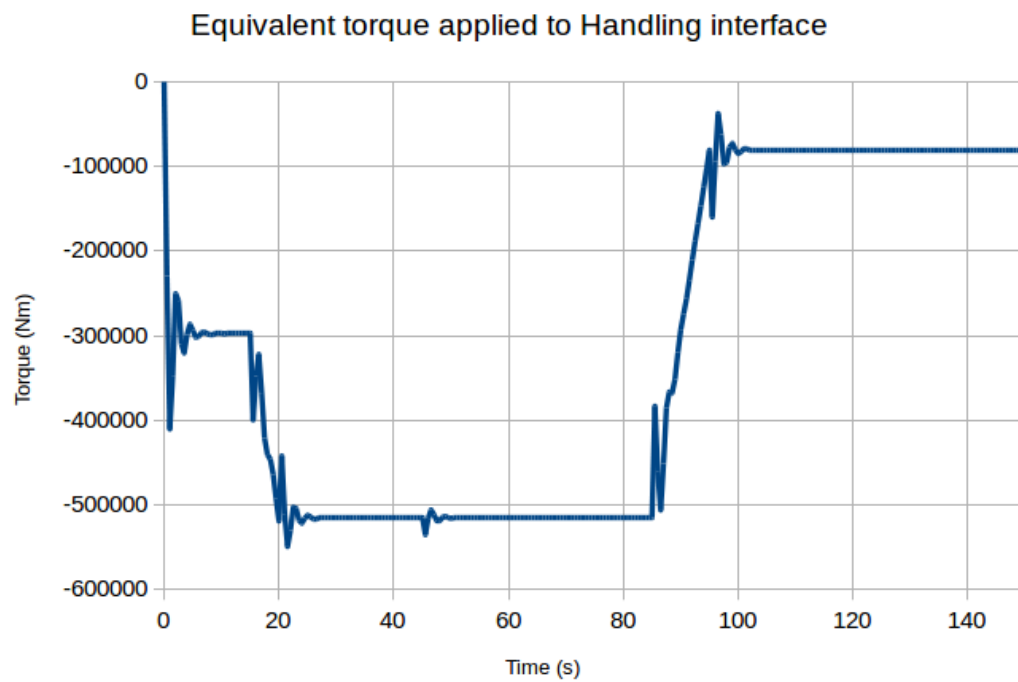


Figure 7.29: Torque applied to the RH interface when removing the MMS

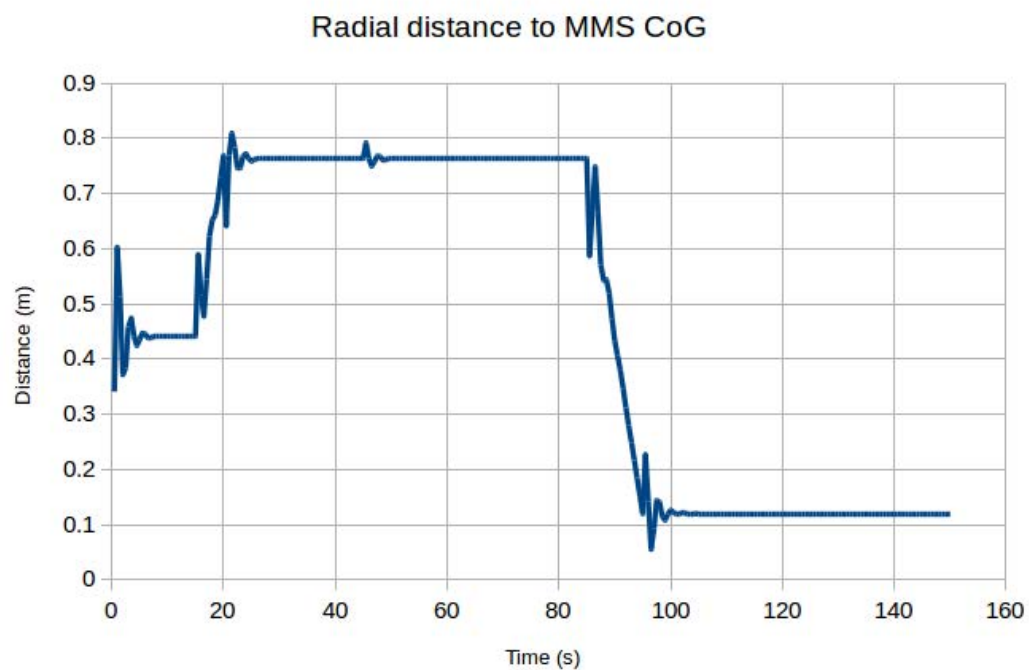


Figure 7.30: Distance from the centre of the RH interface to the Centre of Gravity of the MMS

7.2.5 Computing efficiency

Finally, it's worth detailing the impact of each of the approaches on the computational cost of the simulation. The systems are modelled with fully nonlinear material behaviour, and the analysis use implicit integration schemes. In case of using a model with more nodes than needed, the impact on the computational resources to solve it can be very important.

A. 2D analyses

In the 2D case, the EFG and FEM results have been presented. In order to achieve enough accuracy with FEM, a quite dense mesh has been defined. The main effect on the computational expense is that the system no longer is solved in less time than the simulated time (Figure 7.31). In fact, the time required for solving the analysis is doubled with respect to the time analysed. On contrary, the EFG model is solved always well below the time step size, although step solving time is more sensitive to nonlinearities (in average).

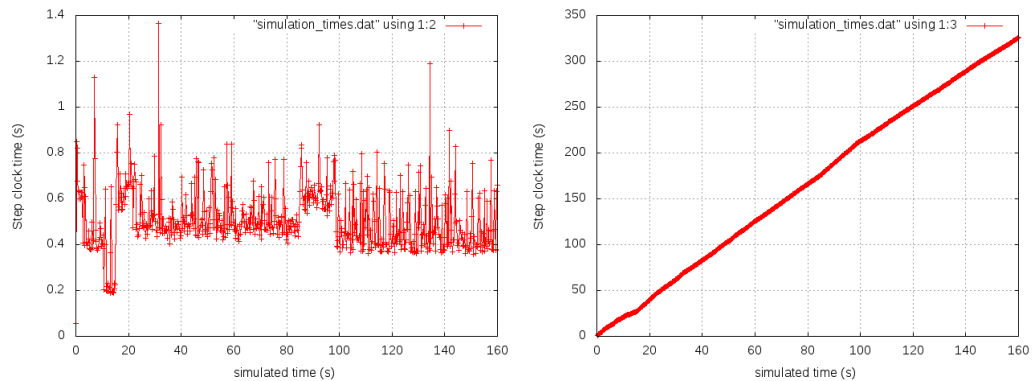


Figure 7.31: Computational times for the 2D FEM model. Time per 0.25 s step (left), and accumulated clock time (right)

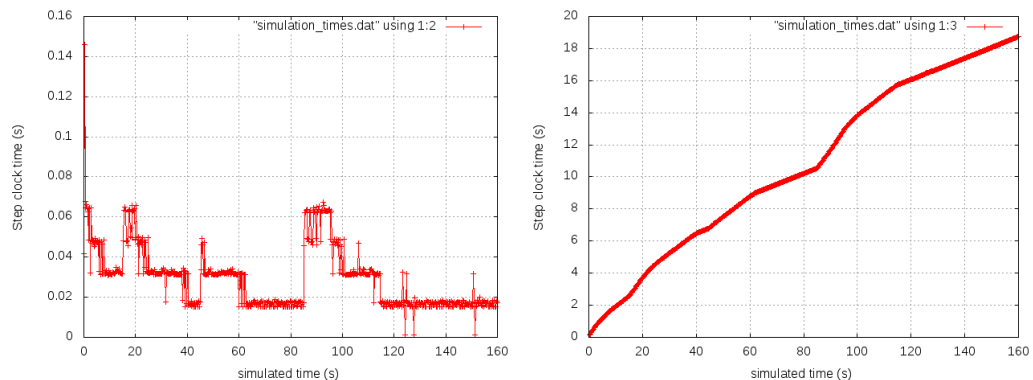


Figure 7.32: Computational times for the 2D EFG model. Time per 0.25 s step (left), and accumulated clock time (right)

B. 3D analyses

For the 3D models, sparse matrices are used, although the main impact in computational cost is the iteration through the support nodes to compute forces and system's matrices. Special attention has been paid to optimizing the calculation of the tangent matrix, adding also a shared memory parallel execution routine to take advantage of multi-core processors.

Simulation speeds are encouraging in the case of the coarse model (7.33). A lot of steps are computed well under the step size, while the maximum duration is within 4 times the step size. This results can clearly be improved for a final application, considering that the system has been studied with a very high flexibility, and that the simulations were run in a regular laptop¹. One drawback of meshfree methods needs to be noted: the exponential need of resources. When the dense node distribution was solved, the increase in clock time needed was very important, leading to analysis runs of around two hours for the minimum support size and GP number.

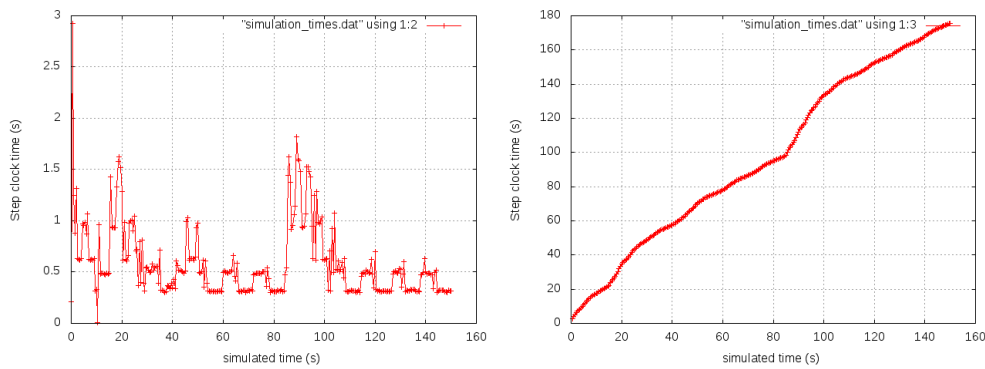


Figure 7.33: Computational times for the 3D EFG coarse model. Time per 0.5 s step (left), and accumulated clock time (right)

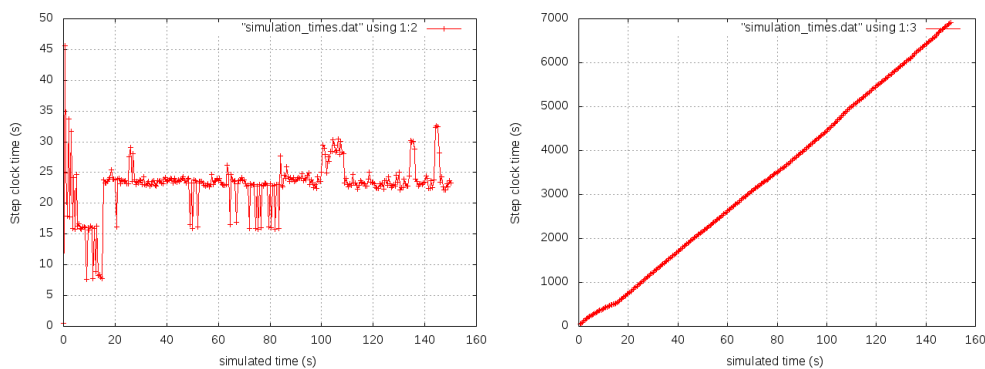


Figure 7.34: Computational times for the 3D EFG dense model. Time per 0.5 s step (left), and accumulated clock time (right)

¹Intel® Core™ i7-4710HQ quad core CPU @ 2.50GHz

Conclusions and Outlook

8

8.1 Overview of main contributions

This thesis has proposed the following set of new contributions to the field of multibody dynamics:

1. The work presents for the first time the use of meshfree methods in multibody dynamics, using a Total langrangian nonlinear formulation for hyperelastic bodies.
2. A novel generalized formulation of constraints and joints is presented, valid for rigid and flexible bodies; and compatible with FEM and meshfree discretizations. This completely new approach allows to define the joint independently of the body nodes without the addition of any more constraints to the system.
3. A new strategy for automatic contacts is proposed based on meshfree neighbour search algorithms, with applications in 2D and extendible also to 3D.
4. Compatible parameters are proposed for defining meshfree bodies either with Radial Basis Functions or Moving Least Squares aproximations.
5. A new methodology is applied to the discretization of bodies. It allows to reduce significantly the size of the problem, using a dense grid for capturing a very detail geometry and a coarser node distribution for modelling the dynamics of the system. It is also shown to work with bodies defined by multiple domains.
6. A complete multibody computer code has been built from scratch, including the algebraic methods [32], linear and nonlinear solvers, time integrators, etc.

for the thermo-mechanical simulation of flexible and rigid bodies. It has been designed in a way that shares as much as possible code between the different methods for modelling the flexibility of the parts: FEM and meshfree. Although its development has required a huge effort, it allows to have a really precise comparison and benchmark of the new approach.

7. The use of meshfree methods in mechanical [32] [35], thermal and thermo-mechanical [34] nonlinear MB dynamics has been benchmarked for the first time. The comparison to FEM shows an order of magnitude better accuracy at equal number of nodes, and general gains of efficiency, specially for systems with coarse discretizations.
8. Examples of all the formulation proposed to apply the meshfree approximation in a MB framework are provided, demonstrating that the approach is generic and extensible to further problems.
9. A final application to a real system has been presented [36], allowing to overcome day-to-day real engineering issues such as poor CAD modelling and non-conformity of meshes, and thus reducing the overall required modelling time. Coarse discrete models of the system have been proven to be capable of Real-time simulation at this time in 2D, and in a near future in 3D.

8.2 Summary of conclusions and research findings

The document has been divided into several chapters, in order to clarify the different building blocks of the research. The first chapter introduces the motivation for looking into alternative formulations for simulating the nonlinear behaviour of systems including rigid and flexible bodies. The kind of applications where issues have been found using the usual FEM approach are centred in the Fusion Technology field; specifically for the simulation of future multibody remote handling equipment, and for present linear particle accelerators. Galerkin meshfree methods are studied for improving the accuracy of the thermo-mechanical behaviour of the flexible bodies, which is the main problem statement. A comprehensive review of the State of the Art shows that no previous application of the meshfree shape functions has been proposed previously. This is believed to completely justify the aims and scope of this PhD. The need for an objective comparison between FEM and Meshfree methods in multibody dynamics has a very important consequence: a complete code including both methods will need to be developed. This is required in order to share as much functions as possible and have a common solver core, as it's the only way to put side by side both their accuracy and efficiency.

The second chapter is devoted to define the formulation framework for nonlinear multibody dynamics of flexible bodies, including thermal effects. The Total

Lagrangian formulation of the equations is chosen to take advantage of the equilibrium being set in the reference configuration. This implies not needing an update of the shape functions during the deformation of the system, which is key to get an efficient implementation of discrete meshfree dynamic systems. A thorough definition of all relevant laws and equations is provided, which leads to the continuous variational formulation for the flexible bodies.

The meshfree shape functions are presented in Chapter 3. Radial Basis Functions (RBF) and Moving Least Squares (MLS) have been selected, as representatives of families of functions with and without the Kronecker delta property. Their definition has been reformulated in order to have equivalent length and interpolation properties; leading to a common set of parameters used to define the approximation of any point within the bodies.

Chapter 4 presents the main theoretical contribution of the thesis. In order to complete the definition of any multibody system, the other components are presented. The configuration of multi-dimensional rigid bodies is described using only cartesian coordinates for their nodes, along with a set of constraints between them. This is known to have the advantage of leading to a constant Mass matrix. The relation between a flexible body's shape function with the rigid bodies' constant matrix is stated, which in fact allows to use signed FEM shape functions to define the configuration of the rigid body. Following the same argument, the meshfree shape functions are used to defined the position of any point of a flexible body either inside or outside its domain. This is used to develop a generalised formulation for mechanical and thermal constraints. Real joints are also detailed in this mesh-free generalized form, which is valid for any kind of discretization of the flexible body's domain, including FEM. A generalised contact formulation is proposed and developed for 2D. It takes advantaged of the same techniques needed for neighbour search when building the shape functions, and it is also expandable to 3D application. Finally, the global system of equations is presented, describing the weakly coupled Total Lagrangian thermo-mechanical form for the dynamic equilibrium of the multibody system.

The rest of the thesis explains how the proposed formulation can be applied. To validate and verify both the equations and their implementation in the complete multibody code developed, three different cases are studied in Chapter 5. From a simple triangle to a thick-wall hollowed beam, the integration point distribution, shape function values, and overall mechanical and thermal response are compared between FEM, RBF and MLS discretizations. Advantages of the proposed approach are identified when dealing with non-conforming meshes and multi-domain bodies.

A large effort has been done to compare the efficiency of FEM and meshfree in multibody benchmarks in Chapter 6. Mechanical simulations with large displace-

ment demonstrate that the meshfree methods are consistently more robust to extreme deformations, and also give more accurate solutions for an equal number of degrees of freedom. For a given simulation time, they also show more efficient as the number of nodes decrease. One of the motivation problems is studied also in this chapter: the thermal simulation of IFMIF LIPAc diagnostic slits. A novel strategy for the definition of the volumetrical thermal load is described, and when used in conjunction with the RBF shape functions, an increase in efficiency of one order of magnitude is found compared with FEM. The final benchmark compares the coupled thermo-mechanical behaviour, and similar conclusions are drawn. A linear thermal load is used in this last case, helping the FEM to capture correctly the peak temperature; although the meshfree methods show a minimum error converging to the reference solution during the diffusion cooling process.

The last chapter is divided in two big sections. The first one presents three examples where meshfree allow to simulate systems that FEM cannot cope with—without requiring any involved extra algorithms. The powerful approach of the generalized joints formulation is demonstrated by a simple pin joint, where the hollow shape of the pin is removed from the flexible component, but retaining the centre of rotation of the joint, even when the boundary deforms. Finally, a real-world application is analysed with the proposed formulation, showing very good comparison with finite elements, and a greater efficiency, which overcomes a radical issue for its application to real-time simulations.

8.3 Future work

From the lessons learnt through the thesis, the following research lines are proposed to continue investigating and extending the meshfree applications in multi-body dynamics:

1. Compare the accuracy and efficiency of other meshfree shape functions, such as Maximum Entropy interpolants.
2. Further extend the contact formulation to take advantage also of the smooth discrete shape of the boundary provided by meshfree functions.
3. Fully meshless methods can be also applied by implementing other integration techniques, such as node collocation with strain smoothing stabilization.
4. Investigate and apply discretization refinement algorithms and specialized integrator schemes (variable step-size, energy-momentum conserving, predictor-corrector, etc) to improve accuracy, robustness, and control of the numerical error during the simulation.

5. Extend the formulation with other nonlinear material models, especially with localized plastic deformation. This study could be done also using local updates in the shape functions when the deformation of a given integration point is excessive.
6. Implement the methods in massively parallel computing systems. As the main bottleneck has been identified to be the computation and assembly of the tangent matrix, adapting the formulation to run in shared memory devices such as GPUs and co-processors can greatly increase their efficiency.
7. Develop methods and algorithms to optimize the shape functions definition: from the neighbour search process, to the optimal number of supporting nodes, and the selection of the kernel function type and parameters.
8. Automatize the process from the CAD modelling stage to the final simulation run. Meshfree methods are extremely well suited for this task, as they pose very little requirements in the discrete integration grid. Also, in case it is necessary to cope with varied geometries, the nodes defining the dynamics of the system can be erased, added, or modified very easily, allowing a straightforward implementation of discretization algorithms. Automating the joint definition between different bodies is also feasible, as it has been demonstrated that the constraints can be formulated independently of the discrete form of the joined bodies.

Bibliography

- [1] N. Akkiraju, H. Edelsbrunner, M. Facello, P. Fu, E. P. Mücke, and C. Varella. Alpha Shapes: Definition and Software. In *Proceedings of the 1st International Computational Geometry Software Workshop*, pages 63–66, 1995.
- [2] F. Amirouche. *Fundamentals of Multibody Dynamics. Theory and Applications*. Birkhäuser Boston, 2006.
- [3] M. Arroyo and M. Ortiz. Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods. *International Journal for Numerical Methods in Engineering*, 65(13):2167–2202, 2006.
- [4] C. Bachmann, G. Aiello, R. Albanese, R. Ambrosino, F. Arbeiter, J. Aubert, L. Boccaccini, D. Carloni, G. Federici, U. Fischer, M. Kovari, A. Li Puma, A. Loving, I. Maione, M. Mattei, G. Mazzone, B. Meszaros, I. Palermo, P. Pereslavitsev, V. Riccardo, P. Sardain, N. Taylor, S. Villari, Z. Vizvary, A. Vaccaro, E. Visca, and R. Wenninger. Initial {DEMO} tokamak design configuration studies. *Fusion Engineering and Design*, 98–99:1423–1426, 2015. Proceedings of the 28th Symposium On Fusion Technology (SOFT-28).
- [5] K.J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice Hall Inc, New Jersey, 1982.
- [6] T. Belytschko and M. Fleming. Smoothing, enrichment and contact in the element-free Galerkin method. *Computers & Structures*, 71(2):173–195, Apr. 1999.
- [7] T. Belytschko, Y. Krongauz, M. Fleming, D. Organ, and W.K. Liu. Smoothing and accelerated computations in the element free Galerkin method. *Journal of Computational and Applied Mathematics*, 74(1):111–126, Nov. 1996.

- [8] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(1):3–47, Dec. 1996.
- [9] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2):229–256, 1994.
- [10] B. Brañas, D. Iglesias, F. Arranz, G. Barrera, N. Casal, M. García, J. Gómez, D. López, J.I. Martínez, F. Martín-Fuertes, F. Ogando, C. Oliver, J. Sanz, P. Sauvan, and A. Ibarra. Design of a beam dump for the IFMIF-EVEDA accelerator. *Fusion Engineering and Design*, 84(2–6):509–513, 2009. Proceeding of the 25th Symposium on Fusion Technology(SOFT-25).
- [11] F. Brezzi. On the Existence, Uniqueness and Approximation of Saddle-point Problems Arising from Lagrangian Multipliers. *R.A.I.R.O.*, 8:129–151, 1974.
- [12] J.-S. Chen. Reproducing Kernel Particle Methods for large deformation analysis of non-linear structures. *Computer Methods in Applied Mechanics and Engineering*, 139:195–227, December 1996.
- [13] J. S. Chen, V. Kotta, H. Lu, D. Wang, D. Moldovan, and D. Wolf. A variational formulation and a double-grid method for meso-scale modeling of stressed grain growth in polycrystalline materials. *Computer Methods in Applied Mechanics and Engineering*, 193(12):1277–1303, Mar. 2004.
- [14] J.-S. Chen, D. Wang, and S. B. Dong. An extended meshfree method for boundary value problems. *Computer Methods in Applied Mechanics and Engineering*, 193(12):1085–1103, Mar. 2004.
- [15] J.-S. Chen and H.-P. Wang. New boundary condition treatments in meshfree computation of contact problems. *Computer Methods in Applied Mechanics and Engineering*, 187(3):441–468, July 2000.
- [16] M. Coleman, N. Sykes, D. Cooper, D. Iglesias, R. Bastow, A. Loving, and J. Harman. Concept for a vertical maintenance remote handling system for multi module blanket segments in {DEMO}. *Fusion Engineering and Design*, 89(9–10):2347–2351, 2014.
- [17] European Commission. Fusion Electricity, November 2012.
- [18] S. Fernández-Méndez, J. Bonet, and A. Huerta. Continuous blending of SPH with finite elements. *Computers & Structures*, 83(17):1448–1458, June 2005.
- [19] R. Franke. Scattered data interpolation: test of some methods. *Math Computation*, 38:181–200, 1982.

- [20] J. Garcia de Jalon and E. Bayo. *Kinematic and Dynamic Simulation of Multibody Systems. The Real Time Challenge*. Springer-Verlag, 1994.
- [21] J. García de Jalón and E. Bayo. *Kinematic and Dynamic Simulation of Multibody Systems. The Real Time Challenge*. Springer-Verlag, New York, USA, 1994.
- [22] J. C. García Orden. *Dinámica no lineal de sistemas multicuerpo flexibles mediante algoritmos conservativos*. PhD thesis, Universidad Politécnica de Madrid, ETSICCP, 1999.
- [23] J. Gerstmayr, A. Dorninger, R. Eder, P. Gruber, D. Reischl, M. Saxinger, M. Schörgenhumer, A. Humer, K. Nachbagauer, A. Pechstein, and Y. Vetyukov. HOTINT - a script language based framework for the simulation of multibody dynamics systems. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. ASME, August 4-7 2013.
- [24] D. González, E. Cueto, and M. Doblaré. A high order method using MAX-ENT approximation schemes. *International Journal of Material Forming*, 2(1):577–580, 2009.
- [25] O. Gonzalez and A. M. Stuart. *Introduction to Continuum Mechanics*. Stanford University, 1995.
- [26] E. Hardee, K.-H. Chang, I. Grindeanu, S. Yoon, M. Kaneko, and J.-S. Chen. A structural nonlinear analysis workspace (SNAW) based on meshless methods. *Adv. Eng. Softw.*, 30(3):153–175, 1999.
- [27] Z. Hehua, L. Wenjun, C. Yongchang, and M. Yuanbin. A meshless local natural neighbor interpolation method for two-dimension incompressible large deformation analysis. *Engineering Analysis with Boundary Elements*, 31(10):856–862, Oct. 2007.
- [28] R. Heidinger, J. Knaster, H. Matsumoto, M. Sugimoto, A. Mosnier, F. Arbeiter, N. Baluc, P. Cara, S. Chel, A. Facco, P. Favuzza, V. Heinzl, A. Ibarra, V. Massaut, G. Micciche, F.S. Nitti, and J. Theile. Progress in {IFMIF} Engineering Validation and Engineering Design Activities. *Fusion Engineering and Design*, 88(6–8):631–634, 2013. Proceedings of the 27th Symposium On Fusion Technology (SOFT-27); Liège, Belgium, September 24-28, 2012.
- [29] D.A. Hu, S.Y. Long, X. Han, and G.Y. Li. A meshless local Petrov–Galerkin method for large deformation contact analysis of elastomers. *Engineering Analysis with Boundary Elements*, 31(7):657–666, July 2007.

- [30] D. Iglesias, F. Arranz, J.M. Arroyo, G. Barrera, B. Brañas, N. Casal, M. García, D. López, J.I. Martínez, A. Mayoral, F. Ogando, M. Parro, C. Oliver, D. Rapisarda, J. Sanz, P. Sauvan, and A. Ibarra. The IFMIF-EVEDA accelerator beam dump design. *Journal of Nuclear Materials*, 417(1–3):1275–1279, 2011.
- [31] D. Iglesias, F. Arranz, M. Parro, D. Rapisarda, B. Brañas, N. Casal, J. M. Carmona, I. Podadera, C. Oliver, and A. Ibarra. Thermo-mechanical Design of Particle-stopping Devices at the High Energy Beamline Sections of the IFMIF/EVEDA Accelerator. In *Proceedings of IPAC2011*, September 2011.
- [32] D. Iglesias and J. J. Arribas. The Linked Matrix methods collection for flexible sequential and parallel object-oriented programming. In C. L. Botasso, P. Massarati, and L. Trainelli, editors, *Multibody Dynamics 2007, ECCOMAS Thematic Conference*, June 2007.
- [33] D. Iglesias, R. Bastow, D. Cooper, R. Crowe, D. Middleton-Gear, R. Sibois, D. Carloni, Z. Vizvary, O. Crofts, J. Harman, and A. Loving. Remote handling assessment of attachment concepts for {DEMO} blanket segments. *Fusion Engineering and Design*, 98–99:1500–1504, 2015.
- [34] D. Iglesias, J. C. García Orden, B. Brañas, J.M. Carmona, and J. Molla. Application of Galerkin meshfree methods to nonlinear thermo-mechanical simulation of solids under extremely high pulsed loading. *Fusion Engineering and Design*, 88(9–10):2744–2747, 2013.
- [35] D. Iglesias and J.C. García Orden. Galerkin meshfree methods applied to the nonlinear dynamics of flexible multibody systems. *Multibody System Dynamics*, 25(2):203–224, 2011.
- [36] D. Iglesias, J. C. García Orden Orden, and A. Loving. Simulating the Remote Handling of the Blanket Segments in Demo Fusion Reactor With Thermo-mechanical Meshfree Multibody Dynamics. In *11th World Congress on Computational Mechanics (WCCM XI)*, July 2014.
- [37] M. Iura and J. Kanaizuka. Flexible translational joint analysis by meshless method. *International Journal of Solids and Structures*, 37(37):5203–5217, Sep. 2000.
- [38] E. Kansa. Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations. *Computers & Mathematics with Applications*, 39:123–137, April 2000.
- [39] E.J. Kansa. Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics - I surface approximations and partial

- derivative estimates. *Computers & Mathematics with Applications*, 19(8):127–145, 1990.
- [40] E.J. Kansa. Exact explicit time integration of hyperbolic partial differential equations with mesh free radial basis functions. *Engineering Analysis with Boundary Elements*, 31(7):577–585, July 2007.
- [41] J. Knaster, S. Chel, U. Fischer, F. Groeschel, R. Heidinger, A. Ibarra, G. Micciche, A. Möslang, M. Sugimoto, and E. Wakai. IFMIF, a fusion relevant neutron source for material irradiation current status. *Journal of Nuclear Materials*, 453(1–3):115–119, 2014.
- [42] Y. Krongauz and T. Belytschko. Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Methods in Applied Mechanics and Engineering*, 131:133–145, April 1996.
- [43] P. Lancaster and K Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158, July 1981.
- [44] J. Li, Y. C. Hon, and C.-S. Chen. Numerical comparisons of two meshless methods using radial basis functions. *Engineering Analysis with Boundary Elements*, 26(3):205–225, Mar. 2002.
- [45] S. Li and W.K. Liu. Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, 55(1):1–34, 2002.
- [46] T. Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers & Structures*, 11(1):83–95, Feb. 1980.
- [47] G. R. Liu. *Mesh Free Methods: Moving Beyond the Finite Element Method*. CRC, 1 edition, July 2002.
- [48] G.R. Liu, G.Y. Zhang, Y.Y. Wang, Z.H. Zhong, G.Y. Li, and X. Han. A nodal integration technique for meshfree radial point interpolation method (NI-RPIM). *International Journal of Solids and Structures*, 44(11):3840–3860, June 2007.
- [49] L. Liu, L.P. Chua, and D.N. Ghista. Element-free Galerkin method for static and dynamic analysis of spatial shell structures. *Journal of Sound and Vibration*, 295(1):388–406, Aug. 2006.
- [50] W.K. Liu, S. Jun, and Y.F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8):1081–1106, 1995.

- [51] X. Liu, G.R. Liu, K.Tai, and K.Y. Lam. Radial point interpolation collocation method (RPICM) for partial differential equations. *Computers & Mathematics with Applications*, 50:1425–1442, Oct.-Nov. 2005.
- [52] S.Y. Long, K.Y. Liu, and D.A. Hu. A new meshless method based on MLPG for elastic dynamic problems. *Engineering Analysis with Boundary Elements*, 31(1):43–48, Jan. 2006.
- [53] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82:1013–1024, dec 1977.
- [54] D. Meagher. *Octree Encoding: a New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer*. Electrical and Systems Engineering Department Rensselaer Polytechnic Institute Image Processing Laboratory, 1980.
- [55] J. J. Monaghan. An introduction to SPH. *Computer Physics Communications*, 48:89–96, jan 1988.
- [56] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.
- [57] V.P. Nguyen, T. Rabczuk, S. Bordas, and M. Duflot. Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation*, 79(3):763–813, 2008.
- [58] O. Nomen, J. I. Martínez, F. Arranz, D. Iglesias, G. Barrera, B. Brañas, F. Ogando, J. Molla, and M. Sanmartí. Detailed mechanical design of the {LI-PAc} beam dump radiological shielding. *Fusion Engineering and Design*, 88(9–10):2723–2727, 2013.
- [59] M. Parro, N. Casal, D. Iglesias, F. Arranz, and B. Brañas. Design and analysis of the IFMIF–EVEDA beam dump cooling system. *Fusion Engineering and Design*, 87(4):332–335, 2012.
- [60] N. Perrone and R. Kao. A general finite difference method for arbitrary meshes. *Computers & Structures*, 5(1):45–57, 1975.
- [61] J.-P. Ponthot and T.Belyschko. Arbitrary Lagrangian-Eulerian formulation for element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 152:19–46, 1998.
- [62] T. Rabczuk and T. Belytschko. A three-dimensional large deformation mesh-free method for arbitrary evolving cracks. *Computer Methods in Applied Mechanics and Engineering*, 196(29):2777–2799, May 2007.

- [63] T. Rabczuk, T. Belytschko, and S. P. Xiao. Stable particle methods based on Lagrangian kernels. *Computer Methods in Applied Mechanics and Engineering*, 193(12):1035–1063, Mar. 2004.
- [64] I. Romero. On the stability and convergence of fully discrete solutions in linear elastodynamics. *Computer Methods in Applied Mechanics and Engineering*, 191(35):3857–3882, 2002.
- [65] M. Schörgenhumer, P.G. Gruber, and J. Gerstmayr. Interaction of flexible multibody systems with fluids analyzed by means of smoothed particle hydrodynamics. *Multibody System Dynamics*, 30(1):53–76, June 2013.
- [66] A.A. Shabana. *Dynamics of Multibody Systems*. Wiley, 1998.
- [67] Hang Si and Klaus Gärtner. Meshing Piecewise Linear Complexes by Constrained Delaunay Tetrahedralizations. In *14th International Meshing Roundtable*, pages 147–164, San Diego, CA, USA, September 2005. Springer-Verlag.
- [68] N. Sukumar. Construction of polygonal interpolants: a maximum entropy approach. *International Journal for Numerical Methods in Engineering*, 61(12):2159–2181, 2004.
- [69] N. Sukumar, B. Moran, and T. Belytschko. The Natural Element Method in Solid Mechanics. *International Journal for Numerical Methods in Engineering*, 43:839–887, 1998.
- [70] N. Sukumar and R. W. Wright. Overview and construction of meshfree basis functions: from moving least squares to entropy approximants. *International Journal for Numerical Methods in Engineering*, 70(2):181–205, 2007.
- [71] D. Wang and J.-S. Chen. Locking-free stabilized conforming nodal integration for meshfree Mindlin–Reissner plate formulation. *Computer Methods in Applied Mechanics and Engineering*, 193(12):1065–1083, Mar. 2004.
- [72] Y. You, J.-S. Chen, and T. E. Voth. Characteristics of semi- and full discretization of stabilized Galerkin meshfree method. *Finite Elements in Analysis and Design*, 38(10):999–1012, Aug. 2002.

Contact forces with varying surface normal

A

A.1 2D case

The computation of the contact forces is defined by the potential detailed in section 4.5. The formulation of the first and second of its gradients when the normal vector is not constant can be calculated as follows:

$$\begin{aligned}\frac{\partial \Phi}{\partial x_0} &= \frac{0.5 (x_1 - x_0) ((x_0 - x_1) (2 y_2 - y_1 - y_0) + (2 x_2 + x_1 - x_0) (y_1 - y_0))^2}{((y_1 - y_0)^2 + (x_1 - x_0)^2)^2} \\ &+ \frac{0.5 (2 y_2 - 2 y_1) ((x_0 - x_1) (2 y_2 - y_1 - y_0) + (2 x_2 + x_1 - x_0) (y_1 - y_0))}{(y_1 - y_0)^2 + (x_1 - x_0)^2}\end{aligned}$$

$$\begin{aligned}\frac{\partial \Phi}{\partial y_0} &= \frac{0.5 (y_1 - y_0) ((x_0 - x_1) (2 y_2 - y_1 - y_0) + (2 x_2 + x_1 - x_0) (y_1 - y_0))^2}{((y_1 - y_0)^2 + (x_1 - x_0)^2)^2} \\ &- \frac{1.0 x_2 ((x_0 - x_1) (2 y_2 - y_1 - y_0) + (2 x_2 + x_1 - x_0) (y_1 - y_0))}{(y_1 - y_0)^2 + (x_1 - x_0)^2}\end{aligned}$$

$$\begin{aligned}\frac{\partial \Phi}{\partial x_1} &= \frac{0.5 (2 y_1 - 2 y_2) ((x_0 - x_1) (2 y_2 - y_1 - y_0) + (2 x_2 + x_1 - x_0) (y_1 - y_0))}{(y_1 - y_0)^2 + (x_1 - x_0)^2} \\ &- \frac{0.5 (x_1 - x_0) ((x_0 - x_1) (2 y_2 - y_1 - y_0) + (2 x_2 + x_1 - x_0) (y_1 - y_0))^2}{((y_1 - y_0)^2 + (x_1 - x_0)^2)^2}\end{aligned}$$

$$\begin{aligned}\frac{\partial \Phi}{\partial y_1} &= \frac{0.5 (2x_2 + 2x_1 - 2x_0) ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))}{(y_1 - y_0)^2 + (x_1 - x_0)^2} \\ &- \frac{0.5 (y_1 - y_0) ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))^2}{((y_1 - y_0)^2 + (x_1 - x_0)^2)^2}\end{aligned}$$

$$\frac{\partial \Phi}{\partial x_2} = \frac{1.0 (y_1 - y_0) ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))}{(y_1 - y_0)^2 + (x_1 - x_0)^2}$$

$$\frac{\partial \Phi}{\partial y_2} = \frac{1.0 (x_0 - x_1) ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))}{(y_1 - y_0)^2 + (x_1 - x_0)^2}$$

$$\begin{aligned}\frac{\partial^2 \Phi}{\partial x_0^2} &= - \frac{0.5 ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))^2}{((y_1 - y_0)^2 + (x_1 - x_0)^2)^2} \\ &+ \frac{2.0 (x_1 - x_0)^2 ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))^2}{((y_1 - y_0)^2 + (x_1 - x_0)^2)^3} \\ &+ \frac{2.0 (x_1 - x_0) (2y_2 - 2y_1) ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))}{((y_1 - y_0)^2 + (x_1 - x_0)^2)^2} \\ &+ \frac{0.5 (2y_2 - 2y_1)^2}{(y_1 - y_0)^2 + (x_1 - x_0)^2}\end{aligned}$$

$$\begin{aligned}\frac{\partial^2 \Phi}{\partial x_0 \partial x_1} &= \frac{0.5 ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))^2}{((y_1 - y_0)^2 + (x_1 - x_0)^2)^2} \\ &- \frac{2.0 (x_1 - x_0)^2 ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))^2}{((y_1 - y_0)^2 + (x_1 - x_0)^2)^3} \\ &- \frac{1.0 (x_1 - x_0) (2y_2 - 2y_1) ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))}{((y_1 - y_0)^2 + (x_1 - x_0)^2)^2} \\ &+ \frac{1.0 (x_1 - x_0) (2y_1 - 2y_2) ((x_0 - x_1) (2y_2 - y_1 - y_0) + (2x_2 + x_1 - x_0) (y_1 - y_0))}{((y_1 - y_0)^2 + (x_1 - x_0)^2)^2} \\ &+ \frac{0.5 (2y_1 - 2y_2) (2y_2 - 2y_1)}{(y_1 - y_0)^2 + (x_1 - x_0)^2}\end{aligned}$$

Verification codes

B

B.1 Mechanical formulation

B.1.1 3-node triangle mechanical static

Code in Python for verification

```
1  #!/usr/bin/env python
2
3  import numpy as np
4  np.set_printoptions(precision=1)
5
6  # Calculate matrices and static response of the 1triangle geometry
7
8  # Material properties
9  density=7850.;
10 young=0.7E6;
11 poisson=0.3;
12
13 # Geometry
14 x1=0.; y1=0.; z1=0.;
15 x2=1.; y2=0.; z2=0.;
16 x3=0.; y3=1.; z3=0.;
17
18 a1A = x2*y3-x3*y2
19 b1A = y2-y3
20 c1A = x3-x2
21
22 a2A = x3*y1-x1*y3
23 b2A = y3-y1
24 c2A = x1-x3
```

```

25
26 a3A = x1*y2-x2*y1
27 b3A = y1-y2
28 c3A = x2-x1
29
30 Area = .5*(a1A+a2A+a3A)
31 print("Area = " + str(Area))
32
33 Bt=np.array(
34 [[b1A, 0., c1A],
35 [0., c1A, b1A],
36 [b2A, 0., c2A],
37 [0, c2A, b2A],
38 [b3A, 0., c3A],
39 [0., c3A, b3A]])
40
41 B=np.transpose(Bt)
42 print("B = ")
43 print B
44
45 kcom = young / (1 - pow(poisson, 2) )
46 D = kcom*np.array(
47 [[1., poisson, 0.],
48 [poisson, 1., 0.],
49 [0., 0, (1.-poisson)/2.]])
50 print("D = ")
51 print D
52
53 K = 1./(4.*Area) * np.dot(np.dot(Bt, D), B)
54
55 print("K = ")
56 print(K)
57
58 # Consistent capacity matrix
59 MA = np.array(
60 [[2., 0., 1., 0., 1., 0.],
61 [0., 2., 0., 1., 0., 1.],
62 [1., 0., 2., 0., 1., 0.],
63 [0., 1., 0., 2., 0., 1.],
64 [1., 0., 1., 0., 2., 0.],
65 [0., 1., 0., 1., 0., 2.]])
66 # Lumped capacity matrix:
67 #MA = np.asarray(
68 #[[4., 0., 0.],
69 #[0., 4., 0.],

```

```

70 #[0., 0., 4.]])
71 M = density*Area/12.*MA
72 print("M = ")
73 print(M)
74
75 gravity=np.array([0,1,0,1,0,1])
76
77 fext=-np.dot(M,gravity)
78 print("Fext = ")
79 print(fext)
80
81 # Static equilibrium, nodes 0 and 2 eliminated:
82 Kff=K[2:4,2:4] # Free DoF K, only last two rows and columns
83 print("Kff = ")
84 print(Kff)
85
86 fextf=fext[2:4] # Free DoF loads
87 print("Fextf = ")
88 print(fextf)
89
90 uf = np.linalg.solve(Kff,fextf)
91
92 Kcf=K[[0,1,4,5]][:,[2,3]]
93 print("Kcf = ")
94 print(Kcf)
95
96 print(fext[[0,1,4,5][:]])
97
98 fextc = fext[[0,1,4,5][:]] - np.dot(Kcf,uf)
99 print("Reactions = ")
100 print(fextc)
101
102 np.set_printoptions(formatter={'float': lambda x: format(x, '6.3E')})
103 print("uf = ")
104 print(uf)
105
106 u=np.array([])
107 u.resize(6)
108 u[2:4]=uf
109 print("u = ")
110 print(u)
111
112 print("Deformation = ")
113 print(np.dot(B,u))
114

```

```

115 print("Stress = ")
116 print(np.dot(np.dot(D,B),u))

```

Output of the previous script:

```

1 Area = 0.5
2 B =
3 [[-1.  0.  1.  0.  0.  0.]
4  [ 0. -1.  0.  0.  0.  1.]
5  [-1. -1.  0.  1.  1.  0.]]
6 D =
7 [[ 769230.8 230769.2  0. ]
8  [ 230769.2 769230.8  0. ]
9  [    0.      0. 269230.8]]
10 K =
11 [[ 519230.8 250000. -384615.4 -134615.4 -134615.4 -115384.6]
12  [ 250000.  519230.8 -115384.6 -134615.4 -134615.4 -384615.4]
13  [-384615.4 -115384.6 384615.4  0.      0.  115384.6]
14  [-134615.4 -134615.4  0.  134615.4 134615.4  0. ]
15  [-134615.4 -134615.4  0.  134615.4 134615.4  0. ]
16  [-115384.6 -384615.4 115384.6  0.      0.  384615.4]]
17 M =
18 [[ 654.2  0.  327.1  0.  327.1  0. ]
19  [  0.  654.2  0.  327.1  0.  327.1]
20  [ 327.1  0.  654.2  0.  327.1  0. ]
21  [  0.  327.1  0.  654.2  0.  327.1]
22  [ 327.1  0.  327.1  0.  654.2  0. ]
23  [  0.  327.1  0.  327.1  0.  654.2]]
24 Fext =
25 [ -0. -1308.3 -0. -1308.3 -0. -1308.3]
26 Kff =
27 [[ 384615.4  0. ]
28  [    0.  134615.4]]
29 Fextf =
30 [ -0. -1308.3]
31 Kcf =
32 [[-384615.4 -134615.4]
33  [-115384.6 -134615.4]
34  [    0.  134615.4]
35  [ 115384.6  0. ]]
36 [ -0. -1308.3 -0. -1308.3]
37 Reactions =
38 [-1308.3 -2616.7 1308.3 -1308.3]
39 uf =
40 [0.000E+00 -9.719E-03]
41 u =

```

```

42 [0.000E+00 0.000E+00 0.000E+00 -9.719E-03 0.000E+00 0.000E+00]
43 Deformation =
44 [0.000E+00 0.000E+00 -9.719E-03]
45 Stress =
46 [0.000E+00 0.000E+00 -2.617E+03]

```

B.2 Thermal formulation

B.2.1 3-node triangle thermal static

Code in Python for verification

```

1
2 #!/usr/bin/env python
3
4 import numpy as np
5 np.set_printoptions(precision=1)
6
7 # Calculate matrices and first step of the 1triangle geometry
8
9 # Material properties
10 density=1750.;
11 capacity=1348.;
12 kappa=224.;
13
14 # Geometry
15 x1=0.; y1=0.; z1=0.;
16 x2=1.; y2=0.; z2=0.;
17 x3=1.; y3=1.; z3=0.;
18 # Equilateral
19 #x1=0.; y1=0.; z1=0.;
20 #x2=1.; y2=0.; z2=0.;
21 #x3=0.5; y3=np.sqrt(3.)/2.; z3=0.;
22
23 a1A = x2*y3-x3*y2
24 b1A = y2-y3
25 c1A = x3-x2
26
27 a2A = x3*y1-x1*y3
28 b2A = y3-y1
29 c2A = x1-x3
30
31 a3A = x1*y2-x2*y1
32 b3A = y1-y2
33 c3A = x2-x1

```

```

34
35 Area = .5*(a1A+a2A+a3A)
36 print("Area = " + str(Area))
37
38 H = kappa/4./Area* np.asarray(
39 [[b1A**2+c1A**2, b1A*b2A+c1A*c2A, b1A*b3A+c1A*c3A],
40 [b1A*b2A+c1A*c2A, b2A**2+c2A**2, b2A*b3A+c2A*c3A],
41 [b1A*b3A+c1A*c3A, b2A*b3A+c2A*c3A, b3A**2+c3A**2]])
42 print("H = ")
43 print(H)
44
45 # Static equilibrium, eq 1 eliminated:
46 Hc=H[1:3,1:3] # Constrained H, only last two rows and columns
47 print("Hc = ")
48 print(Hc)
49
50 # External heat, eq 1 eliminated, Qc
51 Qc = np.asarray([0., 2.5e6])
52
53 # H*T1 = Q
54 T1 = np.linalg.solve(Hc,Qc)
55 print("T1 = ")
56 print(T1)

```

Output of the previous script:

```

1 Area = 0.5
2 H =
3 [[ 112. -112.  0.]
4 [-112. 224. -112.]
5 [  0. -112. 112.]]
6 Hc =
7 [[ 224. -112.]
8 [-112. 112.]]
9 T1 =
10 [ 22321.4 44642.9]

```

B.2.2 Thermal dynamic

Code in Python for verification

```

1 #!/usr/bin/env python
2
3 import numpy as np
4 np.set_printoptions(precision=1)
5

```

```

6  # Calculate matrices and first step of the 1triangle geometry
7
8  # Material properties
9  density=1750.;
10 capacity=1348.;
11 kappa=224.;
12
13 # Geometry
14 x1=0.; y1=0.; z1=0.;
15 x2=1.; y2=0.; z2=0.;
16 x3=1.; y3=1.; z3=0.;
17
18 a1A = x2*y3-x3*y2
19 b1A = y2-y3
20 c1A = x3-x2
21
22 a2A = x3*y1-x1*y3
23 b2A = y3-y1
24 c2A = x1-x3
25
26 a3A = x1*y2-x2*y1
27 b3A = y1-y2
28 c3A = x2-x1
29
30 Area = .5*(a1A+a2A+a3A)
31 print("Area = " + str(Area))
32
33 H = kappa/4./Area* np.asarray(
34     [[b1A**2+c1A**2, b1A*b2A+c1A*c2A, b1A*b3A+c1A*c3A],
35      [b1A*b2A+c1A*c2A, b2A**2+c2A**2, b2A*b3A+c2A*c3A],
36      [b1A*b3A+c1A*c3A, b2A*b3A+c2A*c3A, b3A**2+c3A**2]])
37
38 print("H = ")
39 print(H)
40
41 # Consistent capacity matrix
42 CA = np.asarray(
43     [[2., 1., 1.],
44      [1., 2., 1.],
45      [1., 1., 2.]])
46 # Lumped capacity matrix:
47 #CA = np.asarray(
48     #[[4., 0., 0.],
49      [0., 4., 0.],
50      [0., 0., 4.]])

```

```

51 C = density*Area*capacity/12.*CA
52 print("C = ")
53 print(C)
54
55 # First step integration, BDF-1:
56 #  $T(n+1) = T(n) + h \cdot \dot{T}(n+1)$ 
57 # Integrator step, h
58 h=0.001
59 # External heat, Q
60 Q = np.asarray([0., 0., 2.5e6])
61
62 #  $(C+h \cdot H) \dot{T} = Q$ 
63 Tdot1 = np.linalg.solve(np.add(C, h*H),Q)
64 print("Tdot1 = ")
65 print(Tdot1)
66
67 T1 = h*Tdot1
68 np.set_printoptions(precision=4)
69 print("T1 = ")
70 print(T1)
71
72 # Integration directly in order 0:
73 T01 = np.linalg.solve(1/h*C,Q)
74 print("T01 = ")
75 print(T01) # Same result as T1

```

Output of the previous script:

```

1 Area = 0.5
2 H =
3 [[ 112. -112.  0.]
4 [-112. 224. -112.]
5 [  0. -112. 112.]]
6 C =
7 [[ 196583.3 98291.7 98291.7]
8 [ 98291.7 196583.3 98291.7]
9 [ 98291.7 98291.7 196583.3]]
10 Tdot1 =
11 [-6.4 -6.4 19.1]
12 T1 =
13 [-0.0064 -0.0064 0.0191]
14 T01 =
15 [-0.0064 -0.0064 0.0191]

```