



INTERNATIONAL DOCTORAL
SCHOOL OF THE USC

Umberto Emil
Morelli

PhD Thesis

Efficient computational
strategies for the control
process of continuous casting
machines

Santiago de Compostela, 2022



TESE DE DOUTORAMENTO

**Efficient computational strategies
for the control process of
continuous casting machines**

Umberto Emil Morelli

ESCOLA DE DOUTORAMENTO INTERNACIONAL

PROGRAMA DE DOUTORAMENTO EN MÉTODOS MATEMÁTICOS

E SIMULACIÓN NUMÉRICA EN ENXEÑARÍA E CIENCIAS APLICADAS



SANTIAGO DE COMPOSTELA

ANO 2022

D./Dña. **Umberto Emil Morelli**

Título da tese: Efficient computational strategies for the control process of continuous casting machines

Presento mi tesis, siguiendo el procedimiento adecuado al Reglamento y declaro que:

- 1) La tesis abarca los resultados de la elaboración de mi trabajo.
- 2) De ser el caso, en la tesis se hace referencia a las colaboraciones que tuvo este trabajo.
- 3) Confirmando que la tesis no incurre en ningún tipo de plagio de otros autores ni de trabajos presentados por mí para la obtención de otros títulos.
- 4) La tesis es la versión definitiva presentada para su defensa y coincide la versión impresa con la presentada en formato electrónico.

Y me comprometo a presentar el Compromiso Documental de Supervisión en el caso que el original no esté depositado en la Escuela.

En **Santiago de Compostela, 20 de mayo de 2022.**

Firma electrónica

AUTORIZACIÓN DEL DIRECTOR DE LA TESIS

**Efficient computational strategies for the control
process of continuous casting machines**

D^a. Peregrina Quintela Estévez

INFORMA:

Que la presente tesis, se corresponde con el trabajo realizado por D. Umberto Emil Morelli, bajo mi dirección, y autorizo su presentación, considerando que reúne los requisitos exigidos en el Reglamento de Estudios de Doctorado de la USC, y que como director de esta no incurre en las causas de abstención establecidas en la Ley 40/2015.

De acuerdo con lo indicado en el Reglamento de Estudios de Doctorado, declara también que la presente tesis doctoral es idónea para ser defendida en base a la modalidad de Monográfica con reproducción de publicaciones, en los que la participación del doctorando fue decisiva para su elaboración y las publicaciones se ajustan al Plan de Investigación.

En Santiago de Compostela, 20 de mayo de 2022

AUTORIZACIÓN DEL DIRECTOR DE LA TESIS

**Efficient computational strategies for the control
process of continuous casting machines**

D^a. Patricia Barral Rodiño

INFORMA:

Que la presente tesis, se corresponde con el trabajo realizado por D. Umberto Emil Morelli, bajo mi dirección, y autorizo su presentación, considerando que reúne los requisitos exigidos en el Reglamento de Estudios de Doctorado de la USC, y que como director de esta no incurre en las causas de abstención establecidas en la Ley 40/2015.

De acuerdo con lo indicado en el Reglamento de Estudios de Doctorado, declara también que la presente tesis doctoral es idónea para ser defendida en base a la modalidad de Monográfica con reproducción de publicaciones, en los que la participación del doctorando fue decisiva para su elaboración y las publicaciones se ajustan al Plan de Investigación.

En Santiago de Compostela, 30 de Mayo de 2022

AUTORIZACIÓN DEL DIRECTOR DE LA TESIS

**Efficient computational strategies for the control
process of continuous casting machines**

D. Gianluigi Rozza

INFORMA:

Que la presente tesis, se corresponde con el trabajo realizado por D. Umberto Emil Morelli, bajo mi dirección, y autorizo su presentación, considerando que reúne los requisitos exigidos en el Reglamento de Estudios de Doctorado de la USC, y que como director de esta no incurre en las causas de abstención establecidas en la Ley 40/2015.

De acuerdo con lo indicado en el Reglamento de Estudios de Doctorado, declara también que la presente tesis doctoral es idónea para ser defendida en base a la modalidad de Monográfica con reproducción de publicaciones, en los que la participación del doctorando fue decisiva para su elaboración y las publicaciones se ajustan al Plan de Investigación.

En Trieste, 30 de Mayo de 2022

ACKNOWLEDGEMENT

After this years-long journey, the list of people to thank is long. I will start by thanking my thesis directors Prof. Quintela, Prof. Barral and Prof. Rozza for their constant support along these years, their guidance and kindness. Moreover, I would like to extend my deepest gratitude to all the SISSA Mathlab team for welcoming me and, especially, to Dr. Stabile who helped me constantly and fruitfully all along the project and made this work possible.

This work is part of the ROMSOC project which is a European Industrial Doctorate Program sponsored by the European Commission in the framework of Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 765374. I would like to pay my special regards to all the ROMSOC network, to Prof. Mehrmann who led the group and Dr. Scholz who took care of most of its organization.

All along this PhD, I have been working within the ITMATI team that, under the leadership of its director Prof. Quintela, coordinated all this project. Special thanks to Ariana Fuentes and Sebastián Bouzas who helped me with the administrative and bureaucratic issues that I faced.

This project has been done in cooperation with our industrial partner Danieli & C.Officine Meccaniche SpA. I am extremely grateful to its R&D department and, in particular, to Eng. Marconi, Eng. Conte and Eng. Bianco for their collaboration and their willingness to open us the doors of their facilities and share their knowledge.

Thanks to all mentioned people, I fruitfully gained lots of knowledge and skills. However, in these years, I definitely lost some bone integrity. In two different accidents I broke both my femurs. Hence, I would like to express my deepest thanks to the orthopedic surgery teams of the Ospedale San Paolo of Savona, Italy, and of the Hospital de Conxo in Santiago De Compostela, Spain. Especially, I should thank Dr. Ferrari for being a friend as well as a orthopedics luminary.

Approaching the end of this list, I had the support of so many new and old friends that to mention them individually is not possible. I will only make an exception to thank Pamela for the parallel talks, our garden and her smile.

In conclusion, my deepest gratitude goes to my family. We had plenty of difficulties, but here we are.

RESUMO EN GALEGO

Esta tese realizouse no marco dunha colaboración entre os autores e a empresa Danieli Group dentro do proxecto Europeo ROMSOC. Danieli Group é unha empresa líder no mercado mundial da construción de maquinarias para a produción de aceiro. En concreto, este traballo enfócase en máquinas para a coada continua de aceiro.

A coada continua de aceiro é un proceso durante o que este metal fundido é coado dentro dun molde, xeralmente formado por catro placas, que permiten a entrada e a saída do aceiro. O molde é arrefriado a través de auga a alta presión que corre en tubaxes no interior das placas do molde. O obxectivo do molde é, por unha banda, dar a forma ao aceiro e, por outra, facer unha primeira solidificación, de xeito que, á saída do molde, o aceiro ten unha capa sólida e pode ser guiado por rolos. Nesta segunda fase, o aceiro é arrefriado mediante auga que incide directamente sobre o aceiro ata a completa solidificación.

A parte máis crítica deste proceso é o molde, o corazón do coador. Como se mencionou, nel empeza a solidificación do aceiro e dáse forma á peza. Por tanto, aí prodúcense moitos fenómenos físicos, como o cambio de fase do aceiro, o seu fluxo turbulento, o transporte de inclusións, a transferencia de calor do aceiro cara ao molde e o seu sistema de refrixeración, e tamén as deformacións termo-mecánicas debidas aos cambios de temperatura e ás forzas que se xeran no aceiro e entre o aceiro e o molde.

Todos estes fenómenos fan que no molde poidan acontecer accidentes que reducen a operatividade e a calidade da maquinaria e tamén a seguridade do operador. Por exemplo, é común que o aceiro se pegue a unha placa interior do molde, e nese caso é necesario diminuír rapidamente a velocidade de coada para facer despegar o aceiro. En caso contrario, o aceiro solidificado non podería saír do molde, obstruíndose este, de xeito que o aceiro líquido podería derramarse por riba, o que suporía un grave problema. Tamén, pode acontecer que o arrefriado non sexa suficiente e, en consecuencia, a capa sólida do aceiro tampouco sexa suficientemente espesa durante a segunda fase e rompa, derramando o aceiro líquido.

Para evitar estes problemas e controlar o progreso da coada, é importante ter unha ferramenta para a súa supervisión neste tramo tan crítico do proceso. Pero non é posible facer medidas no interior do molde, onde o aceiro está solidificándose. A día de hoxe, as únicas medidas posibles son obtidas por termopares colocados nas placas do molde, situados nunha superficie paralela á cara quente das placas a uns poucos

centímetros no interior. Ata o de agora, estas medicións de temperaturas foron utilizadas directamente para supervisar o molde. Pero, este método de supervisión é moi empírico, baséase na experiencia do operador e depende da xeometría específica do molde Polo tanto, para poder supervisar mellor a instalación, é necesario para o operador dispoñer de datos que proporcionen máis información sobre o comportamento do molde. O partner industrial desta investigación identificou o fluxo de calor entre molde e aceiro como o dato de interese para a supervisión. De feito, a visualización deste fluxo en tempo real permite a detección rápida de cada imprevisto e unha repentina reacción. Por iso, o obxectivo desta investigación é determinar o fluxo de calor entre o molde e o aceiro.

Para a estimación deste fluxo de calor, dispoñemos das propiedades físicas do molde, da súa xeometría, da temperatura da auga de arrefriado e das medidas que se obteñen dos termopares. Estes danos a temperatura puntual do molde en cada segundo. O obxectivo deste traballo é, a partir destes datos conseguir estimar en tempo real o fluxo de calor entre molde e aceiro.

Nesta tese estudamos este problema no contexto dos problemas inversos. En particular, expoñemos o problema, a través da asimilación de datos, como un problema de control óptimo, buscando o fluxo de calor que minimiza unha distancia entre as temperaturas medidas nos termopares e as calculadas ao resolver o problema directo de transferencia de calor nas placas do molde co fluxo de calor como condición de contorno. Podemos dividir este traballo en tres fases: a modelización do molde, a estimación do fluxo de calor e a redución do custo computacional para conseguir unha estimación en tempo real.

Con respecto á modelización do molde, moitas opcións serían posibles. Podemos modelar todos ou parte dos fenómenos que acontecen no interior do molde e no sistema de arrefriado. Con todo, isto sería computacionalmente moi esixente e dificilmente sería posible cumprir o requisito de tempo real. Polo tanto, eliximos utilizar como dominio só as placas sólidas do molde e modelar a transferencia de calor que nelas acontece. Con esta elección o fluxo de calor que buscamos é unha condición de contorno do noso modelo (en particular, unha condición de tipo Neumann).

A partir desta elección, desenvolvemos unha xerarquía de modelos do molde. Nesta xerarquía, o primeiro modelo é o máis complexo e, a partir diste, agregando hipóteses simplificadoras, creamos a xerarquía completa. De todos os modelos que conforman, utilizamos dous nas fases sucesivas. Ambos son modelos de conducción de calor nun sólido homoxéneo (as placas do molde), pero un é estacionario e outro non estacionario. En concreto, na Parte I da tese consideramos o caso estacionario e na Parte II o non estacionario.

Como xa mencionamos, con respecto á estimación do fluxo de calor, encadramos o problema no marco da asimilación de datos, explotando a estrutura do control óptimo. Neste marco, moitas metodoloxías foron desenvoltas na literatura. Na primeira parte da tese, testamos o comportamento destes métodos clásicos na solución do noso problema.

Para o test dos métodos desenvolvidos, deseñamos uns casos benchmark. En particular, deseñamos bechmarks académicos e industriais. Os primeiros son casos particulares nos cales a xeometría e a condicións ao bordo permiten de obter unha solución analítica do problema. Entón, permítennos de saber a solución exacta do problema directo e comparala coa calculada. Tamén nestes benchmarks, posicionamos termopares virtuais e aplicámoslle a temperatura exacta da solución analítica como medida.

Polo outro lado, os benchmark industriais son deseñados para parecerse á situación real que se atopa nun colador. Por tanto, a xeometría, os parámetros físicos e as condicións ao bordo son as dun verdadeiro molde. Nestes casos non temos a solución analítica do problema directo e podemos usalos só para testar a estimación do fluxo de calor. En particular, elixímonos arbitrariamente un fluxo de calor e calculamos numericamente o correspondente campo de temperatura. As temperaturas calculadas o onde posicionamos os termopares é usada como medida e así testamos a capacidade dos métodos desenvolvidos de estimar o fluxo que eliximos.

A través destes benchmarks, comprobamos que os métodos clásicos non permiten obter boas estimacións do fluxo de calor. Isto é principalmente debido á proximidade das medidas coa fronteira onde estamos a buscar o fluxo, de xeito que este resulta sobreestimado na proximidade das proxeccións dos termopares na fronteira e subestimado nas áreas entre eles.

Como consecuencia, plantexámonos desenvolver un novo método que poida velar por unha boa estimación do fluxo de calor. No desenvolvemento deste novo método, tivemos en conta que a proxección dos termopares na interface entre molde e aceiro crea unha malla uniforme. A partir de aí, estudamos unha parametrización do fluxo baseada en funcións radiais, cada unha delas centrada na proxección dun termopar. Esta parametrización permítenos reducir a procura nun espazo infinito dimensional a outro finito dimensional. Ademais, puidemos desenvolver un método para a solución do problema de control óptimo que sexa directo e non iterativo (como a maioría dos métodos clásicos). O desenvolvemento deste método directo pasa pola escritura vectorial do funcional custo que queremos minimizar no problema de control óptimo, e a súa derivación con respecto aos pesos da parametrización. Como resultado obtemos que os pesos da parametrización, correspondentes a un punto crítico

do noso funcional, poden atoparse resolvendo un sistema linear que ten o tamaño da parametrización (da orde de poucos centos). Para ensamblar o sistema linear é necesario facer moitas simulacións cuxo custo computacional depende do tamaño da malla elixida. Con todo, podemos facer todas estas simulacións nunha fase offline porque non requiren ningunha medida dos termopares. Entón, neste método explotamos unha división offline-online na cal temos unha fase offline, que é computacionalmente custosa, e unha fase online, que é moi rápida e computacionalmente moi eficiente. Este, no caso de ter un modelo do molde estacionario (Part I) permítenos obter unha estimación do fluxo de calor en tempos moi rápidos que cumpren co requisito de tempo real do problema.

A aplicación deste novo método nos casos benchmark que deseñamos comprobou a súa excelente capacidade de estimar o fluxo de calor. Ademais, aplicando técnicas de regularización, este método é capaz de proporcionar boas estimacións tamén cando as medidas son afectadas por ruído, como é habitual en medidas industriais.

Resumindo, na primeira parte desta tese, empezamos desenvolvendo unha xerarquía de modelos para a modelización do fluxo de calor nos moldes. A continuación, para o modelo estacionario de conducción de calor, expoñemos o problema da estimación do fluxo de calor. Para a solución deste problema aplicamos métodos tradicionais, obtendo unha estimación pobre do fluxo de interese. A continuación, desenvolvemos un novo método que, explotando unha parametrización do fluxo de calor, permítenos obter unha boa estimación e, sendo un método directo, permítenos satisfacer o requisito de tempo real que nos require a aplicación.

Na segunda parte centrámonos no caso non estacionario. O modelo do molde é, tamén esta vez, un modelo de conducción nun sólido homoxéneo, pero considerando os efectos non estacionarios. Neste marco, a formulación do problema de asimilación de datos non é trivial. De feito, como para formular o problema inverso temos que considerar o obxectivo de estimar o fluxo de calor en tempo real, necesitamos estar sempre á fronte da liña temporal en canto ela estírase. Dado que os termopares miden a frecuencia cada Hertz, podemos dividir a liña temporal en anacos dun segundo e expor o problema de estimación no último destes anacos, asumindo que xa calculamos o fluxo de calor para os anacos anteriores. Ao facer isto, creamos unha secuencia de problemas inversos que son resoltos cada un no seu tempo. Para resolver estes problemas inversos, desenvolvemos uns métodos novos que explotan a mesma parametrización espacial que empregamos para o caso estacionario. Pero, no caso non estacionario, necesitamos considerar a variación temporal do fluxo de calor. Para facer isto, establecemos pesos da parametrización non estacionarios. En particular, consideramos dous casos: pesos constantes a anacos e pesos lineares a anacos

en cada intervalo dun segundo. En ámbolos dous casos, conseguimos desenvolver métodos directos para a solución dos problemas inversos.

Para testar estes novos métodos e a súa capacidade para estimar o fluxo de calor deseñamos varios casos benchmark. Os resultados móstrannos que os dous métodos (constante a anacos e linear a anacos) teñen comportamentos moi distintos. No caso da aproximación constante a anacos do fluxo de calor, o método proporciona unha boa estimación e a calidade desta estimación non depende da discretización temporal ou espacial. Ademais, o método é resistente dalgunha forma ao ruído nas medidas.

O método linear a anacos, en cambio, vése moi influenciado pola elección da malla e da discretización temporal. En particular, atopamos que discretizacións máis grosas implican unha mellor estimación do fluxo. E, se usamos unha malla e/o un paso temporal demasiado longo, o método vólvese moi inestable. Ademais, esta inestabilidade atópase tamén na resistencia ao ruído nas medidas. En efecto, este método é máis sensible co anterior.

Na parte I desta tese, dividimos o problema en modelización do molde, solución do problema de estimación do fluxo de calor e redución do custo computacional. Vimos como, no caso estacionario, desenvolver un método directo para a estimación do fluxo xa nos permitiu obter unha solución en tempo real. Por tanto, non foi necesario estudar técnicas para a redución do custo computacional.

Estes novos métodos non estacionarios, en cambio, necesitaron un estudo adicional. Estes requiren en cada intervalo temporal a solución dun problema de transferencia de calor no dominio computacional. Isto significa que, en cada intervalo, é preciso calcular unha solución que require un custo computacional que depende da malla e da discretización espacial utilizada. Polo tanto, estes non garanten un rendemento en tempo real. Para conseguilo, é necesario aplicar técnicas de redución de orde.

A redución de orde é unha das novas fronteiras no campo da análise numérica. A idea de base é utilizar as solucións obtidas a través de simulacións anteriores dun modelo para diferentes valores dun parámetro, para obter novas simulacións a un custo computacional moi inferior. Entre todos os posibles métodos de redución de orde (e.g. neural networks, balancing methods ou nonlinear manifold methods) eliximos traballar con métodos de bases reducidas e, en particular, co método POD-Galerkin. Simplificando, podemos resumir este método como segue: Imaxinemos que temos un problema definido a partir dunha ecuación en derivadas parciais parametrizada e que queremos obter de forma rápida a solución deste problema para distintos valores dun parámetro; dito parámetro pode ser unha propiedade física (e.g. a difusividade), unha condición de contorno, etc. Resolvemos o problema para

distintos valores do parámetro, asumindo que as solucións viven nunha variedade suficientemente regular. Podemos asumir que as solucións do problema para valores do parámetro diferentes dos utilizados pode atoparse a través dunha combinación linear das solucións xa calculadas. Entón, faise unha descomposición ortogonal propia (POD) das solucións xa calculadas, atopando así os modos. Estes modos son ordenados de forma xerárquica, no sentido que os primeiros son os que manteñen máis información. A continuación, faise unha proxección de Galerkin das ecuacións que queremos resolver sobre os primeiros modos e, desta maneira, obtense o modelo reducido.

O modelo que nos ocupa neste traballo ten como parámetro a temperatura inicial do molde ao inicio de cada intervalo temporal. Esta temperatura pode ser calquera, polo tanto, é imposible facer unha mostraxe apropiada. Así que, para esta aplicación, desenvolvemos unha técnica nova para a creación do modelo de orde reducida. Esta nova técnica crea as bases POD de maneira automática durante as iteracións do problema inverso. En particular, en cada intervalo temos unha nova temperatura inicial como dato de entrada para o problema que queremos reducir. Esta temperatura inicial corresponde á solución do problema no anaco de tempo considerado. Entón, se supoñemos que temos dispoñibles durante a fase offline as medidas de temperaturas que veñen dunha coada anterior, podemos utilizar as correspondentes solucións coma se fose o noso mostreo e a partir destas mostraxas calcular os modos POD. Con todo, este proceso requiriría, para os cálculos dos modos, uns cálculos moi onerosos ao final desta fase e unha memoria moi grande para ter almacenadas todas as solucións e todos os modos.

Por iso, investigamos a utilización de técnicas POD incrementais que nos permitan calcular os modos POD sen gardar todas as solucións en memoria. Para facer que este método sexa fiable, é necesario garantir que os modos utilizados sexan bos representantes do novo campo de temperatura inicial que está a entrar a cada paso. Para iso, desenvolvemos un estimador do erro de proxección deste campo de temperaturas sobre os modos. Grazas a este estimador, podemos avaliar se o modelo reducido pode ser utilizado ou se necesita ser expandido a través dunha nova simulación completa. Este novo método para a creación automatizada e data-driven do modelo reducido foi testado nos benchmarks que desenvolvemos e mostrou a súa habilidade en crear un modelo reducido rápido e fiable a través do estimador do erro.

Por último, analizamos a sensibilidade do problema á elección do tamaño da malla e do paso de tempo para o caso de aproximación do fluxo de calor a través de bases lineares en tempo. O algoritmo que estudamos utiliza unha distancia entre as temperaturas calculadas e as medidas para elixir automaticamente unha combinación

de malla e paso de tempo que corresponda a un método de aproximación do fluxo que sexa estable. Tamén este algoritmo demostrou nas probas numéricas ter un bo comportamento.

En resumo, nesta tese nos plantexamos o problema de desenvolver ferramentas matemáticas para o control dun molde para a coada continua de aceiro. Primeiro, traballando co equipo da empresa que colaborou nesta investigación, identificamos que o fluxo de calor entre molde e aceiro é a cantidade de interese para esta aplicación. Por tanto, empezamos identificado como posible formulación un enfoque determinístico no marco da asimilación de datos, de control óptimo e dos problemas inversos. O primeiro paso foi o modelar os fenómenos físicos que ocorren nun molde. Neste respecto, desenvolvemos unha xerarquía de modelos que a partir dun primeiro modelo moi complexo crea novos modelos agregando hipóteses simplificadoras. Esta xerarquía permitiunos ter moi claro as hipóteses que levaba cada modelo á hora de elixilo. Aínda que esta xerarquía inclúe moitos modelos, para o problema seguinte da estimación do fluxo de calor só consideramos dous. ámbolos dous son modelos de conducción de calor na rexión sólida das placas do molde. Pero, un é estacionario e o outro non.

Na primeira parte desta tese enfrontamos o problema de estimación do fluxo de calor no caso estacionario. Primeiro, aplicamos metodoloxías estándares para a solución do problema sen obter resultados satisfactorios. Entón, desenvolvemos un novo método que explota a parametrización do fluxo de calor a través de funcións de base radiais. O resultante método permite unha estimación directa do fluxo de calor e, ademais, ten unha descomposición offline-en liña que permite de ter unha fase en liña moi económica computacionalmente que nos permite a estimación do fluxo en tempo real.

Despois do caso estacionario, movémosnos ao non estacionario. Neste caso, despois de expor o problema inverso de forma secuencial, desenvolvemos uns novos métodos para a estimación dun fluxo de calor non estacionario. Estes explotan a mesma parametrización espacial do caso estacionario, pero utilizan pesos para a parametrización que son tempo dependentes. Estes novos métodos, tamén teñen unha descomposición offline-en liña. Pero, na fase en liña necesitan de resolver un problema cuxo custo computacional depende da discretización utilizada. Entón, non podían ser utilizados tal como estaban e foi necesario utilizar técnicas de redución de orde para cumprir coa petición de tempo real. Entón, explotamos un método de bases reducidas chamado POD-Galerkin para a creación do modelo reducido. Con todo, non foi posible utilizar o método directamente e desenvolvémonos un novo método para a creación automática e data-driven do espazo POD e, agregándoo cun novo

estimador do erro, fixemos este novo algoritmo fiable e resistente.

CONTENTS

Resumo en galego	xiii
1 Introduction and Motivation	1
1.1 Continuous Casting	2
1.2 Control of the Mold	4
1.3 Physical Problem	6
1.3.1 State of the Art	7
1.3.2 Specific Physical Problem	10
1.4 Estimating the Mold-Slab Heat Flux	13
1.4.1 State of the Art	14
1.5 Objectives	17
1.6 Thesis Outline	18
I Estimation of the Boundary Heat Flux by Using a Steady-State Mold Model	19
2 Steady-State Direct Problem	21
2.1 Computational Domain and Notation	21
2.2 Physical Data	24
2.3 General Assumptions for CC Mold Models	25
2.4 Three-Dimensional Steady-State Liquid-Solid Heat Transfer Model	26
2.4.1 Incompressible Navier-Stokes Flow Model	26
2.4.2 Pipe Flow Model	27
2.4.3 Thermal Model	29
2.5 Three-Dimensional Steady-State Heat Conduction Model Coupled with 1D Cooling Model	34
2.6 Three-Dimensional Steady-State Heat Conduction Model	36
2.6.1 Numerical Discretization	37
3 Steady-State Inverse Problem	43
3.1 Inverse Three-Dimensional Steady-State Heat Conduction Problem .	43
3.1.1 Alifanov's Regularization	46

3.1.2	Parameterization of the Boundary Conditions	48
3.2	Inverse Three-Dimensional Steady-State Heat Conduction Problem with Total Heat Flux Measure	52
3.2.1	Alifanov's Regularization	54
3.2.2	Parameterization of the Boundary Conditions	55
4	Benchmarks	59
4.1	Benchmark 1	59
4.1.1	Direct Problem	61
4.1.2	Inverse Problem with Temperature Measurements	62
4.1.3	Inverse Problem with Temperature and Total Heat Flux Mea- surements	73
4.1.4	Conclusions	76
4.2	Benchmark 2	77
4.2.1	Inverse Problem with Temperature Measurements	78
4.2.2	Inverse Problem with Temperature and Total Heat Flux Mea- surements	80
4.2.3	Conclusions	85
5	Conclusions	87
 II Estimation of the Boundary Heat Flux by Using an Unsteady- State Mold Model		89
6	Unsteady-State Direct Problem	91
6.1	Three-Dimensional Unsteady-State Heat Conduction Model	91
6.1.1	Numerical Discretization	93
7	Unsteady-State Inverse Problem	97
7.1	Inverse Three-Dimensional Unsteady-State Heat Conduction Problem	97
7.2	Inverse Solver for S_1^k	99
7.2.1	Piecewise Constant Approximation of the Heat Flux	101
7.2.2	Piecewise Linear Heat Flux	105
7.3	Inverse Solver for S_2^k	108
7.4	Regularization	113
7.5	Discretization Selection Algorithm	115

8	Model Order Reduction for the Speed-up of Inverse Solvers	119
8.1	State of the Art	119
8.2	RB POD-Galerkin Reduced Order Model	121
8.2.1	Reduced Basis Space	121
8.2.2	Reduced Order Model	123
8.3	Incremental Construction of the Reduced Basis Space	124
8.3.1	Incremental Proper Orthogonal Decomposition Algorithm	125
8.3.2	Projection Error Estimator	127
8.3.3	Incremental Algorithm for the Computation of the Reduced Basis	130
8.4	Reduced Order Inverse Solvers	131
9	Benchmarks	135
9.1	Benchmark 1	135
9.2	Benchmark 2	138
9.2.1	Setup of the Test Case	138
9.2.2	Effect of Time and Space Discretization Refinement	140
9.2.3	Effect of Cost Functional Parameter, p_g	142
9.3	Benchmark 3	147
9.3.1	Comparison Between Full and Reduced Order Inverse Solvers	148
9.3.2	Effects of Measurements Noise and Regularization	150
9.3.3	Effect of Cost Functional Parameter, p_g	152
10	Conclusions and Future Prospective	157
A	Annex	181
	Index	188

Acronyms

BC Boundary Condition

CC Continuous Casting

CGM Conjugate Gradient Method

CPU Central Processing Unit

DEIM Discrete Empirical Interpolation Method

DP Discrepancy Principle

EID European Industrial Doctorate

FOM Full Order Model

FVM Finite Volume Method

HTC Heat Transfer Coefficient

IC Initial Condition

IHTP Inverse Heat Transfer Problem

IID Independent and Identically Distributed

LES Large Eddy Simulation

LMM Levenberg-Marquardt Method

MOR Model Order reduction

MSCA Marie Sklodowska-Curie Actions

NN Neural Network

PDE Partial Differential Equation

POD Proper Orthogonal Decomposition

PPDE Parameterized Partial Differential Equation

RANS Reynolds-Averaged Navier-Stokes

RB Reduced Basis

RBF Radial Basis Function

ROM Reduced Order Model

ROMSOC Reduced Order Modelling, Simulation and Optimization of Coupled systems

SEN Submerged Entry Nozzle

SISSA Scuola Internazionale Superiore di Studi Avanzati

SVD Singular Values Decomposition

TSVD Truncated Singular Value Decomposition

1 INTRODUCTION AND MOTIVATION

The present thesis is one of the eleven that are part of the Reduced Order Modelling, Simulation and Optimization of Coupled systems (ROMSOC) - H2020 - Marie Skłodowska-Curie Actions (MSCA) - ITN - 765374 program [137]. ROMSOC is a European Industrial Doctorate (EID) project sponsored by the European Commission in the framework of Horizon 2020 as part of the MSCA [51].

This program brings together fifteen international academic institutions and 11 industry partners to work together on complex industrial problems. These industrial applications investigate very different topics, ranging from the European Extremely Large Telescope optics [145] to fluid-structure interaction in blood pumps [92]. Moreover, the objective of this program has been, not only to solve the industrial problems, but also to use them to boost the development of novel mathematical methodologies with the advantage of testing them in their real world environment.

Being part of the ROMSOC program, the present thesis has been developed in cooperation with the Scuola Internazionale Superiore di Studi Avanzati (SISSA) of Trieste [139], in particular the SISSA mathLab [138], and Danieli & C. Officine Meccaniche SpA of Buttrio [35], both of them being based in Italy. Since Danieli is a worldwide leader in the production of steel plants, the subject of this thesis is related to the steel making. In particular, we deal with the Continuous Casting (CC) of steel focusing on the mold region.

We start this introduction by describing the setting of this research with an overview of the industrial process of steel CC. Then, we describe in Section 1.2 the general problem of controlling the mold behaviour during casting that will serve us as motivation for this investigation. Later, in Section 1.3, we discuss the physical phenomena happening in the mold region, we give an overview on previous attempts to model the mold and introduce the mold models that we use later in the thesis. In Section 1.4, we introduce the problem of estimating the mold-steel heat flux. Since this heat flux is the quantity of interest in controlling the mold behaviour, the mold-steel heat flux estimation will be the core topic of the present thesis. To conclude, we summarize in Section 1.5 the objectives and novelties of this research while Section 1.6 provides the thesis outline.

1.1 CONTINUOUS CASTING

The steel CC is presently the most used process to produce steel worldwide. For example, in 2017, 96% of the steel was produced using CC (see [165]). Continuous casters as in Figure 1.1.1 have been used for many decades now. Consequently, the process has undergone a long sequence of improvements driven by the experience of the commercial operators and, more recently, numerical simulations (see [154]). In this section we provide a general description of the CC process, we refer the interested reader to Irving's monograph [72] for a detailed description.

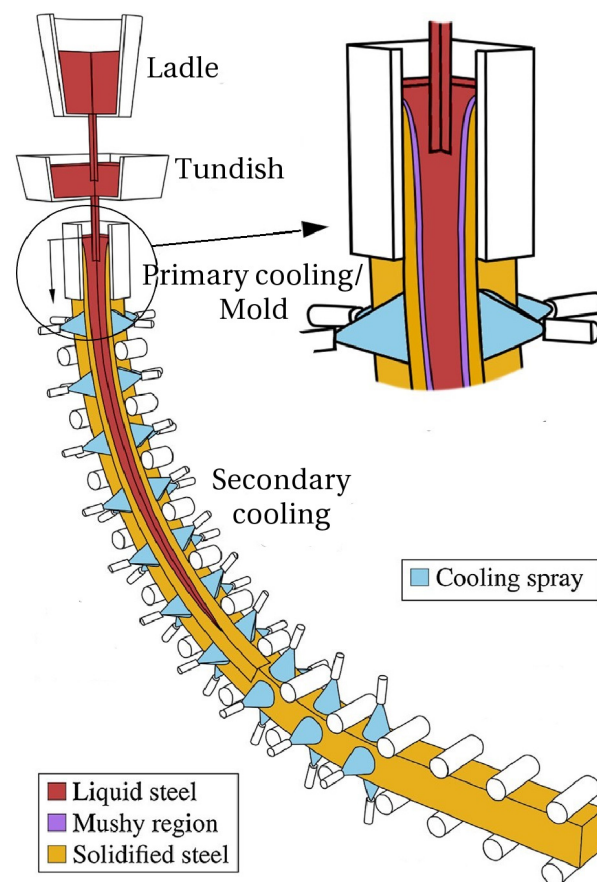


Figure 1.1.1: Schematic overview of the continuous casting process (adapted from [81]). The molten steel is tapped from the ladle to the tundish. Then, the SEN injects it into the mold where the steel begins its solidification. Starting from the mold, the steel slab is cooled and it reaches complete solidification in the secondary cooling region.

We can summarize the CC process as follows. At first, the metal is liquefied. Then, it is tapped into the ladle. When it is at the correct temperature, the metal goes into the tundish. In the tundish, the metal flow is regulated and smoothed. Moreover, the tundish plays the role of a reservoir to continue the casting when the ladle is

empty and must be substituted.

Through the Submerged Entry Nozzle (SEN), the metal is drained into a mold. The SEN drains the metal below a layer of flux powder which is floating at the top of the molten metal (see Figure 1.1.2). Part of the powder in contact with the steel melts down creating a liquid layer. This layer fills the gap between the steel and the mold protecting the steel from re-oxidation. Moreover, it absorbs the non-metallic inclusions, lubricates the steel shell as it passes through the mold, and controls the heat flux between the solidifying steel shell and the mold (see [77]).

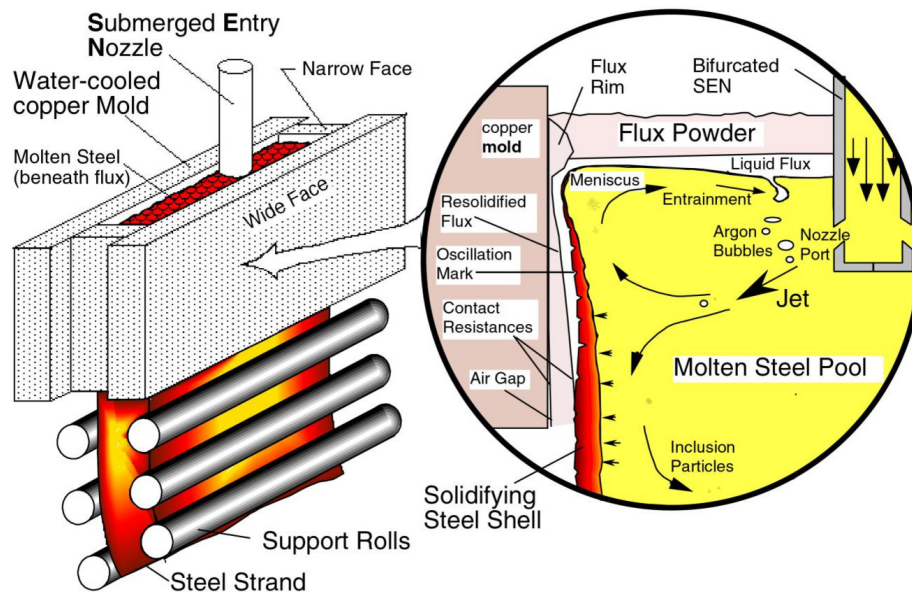


Figure 1.1.2: Schematic of a vertical section of the mold region (taken from [155]). On the left, the main phenomena occurring within the mold are illustrated. In particular, the figure shows the gap which forms between the steel and the mold. This gap is filled by the flux powder which liquefies in contact with the hot steel and resolidifies close to the mold face. In the lower part of the mold an air gap can occur due to the slab shrinkage. Molds are generally tapered to reduce this air gap.

The role of the mold in the CC process is to cool down the steel until it has a solid skin which is thick and cool enough to be supported by rollers. We call slab the mold output. The steel of the slab is still molten in its inner region, thus a secondary cooling section follows the mold. Supported by cooled rollers, the slab is cooled until complete solidification by directly spraying water over it. At the end of this cooling region, the casting is completed. To be ready for its final application, the strand generally continues through additional mechanisms which may flatten, roll or extrude the metal into its final shape.

In this work, we focus on CC of thin slabs. Slabs are cataloged thin when their

thickness is smaller than 70 mm, while their width is between 1 and 1.5 m, in general. Thanks to the small thickness, the solidification in the slab is relatively fast, consequently the casting speed is generally high, between 7 and 14 meters per minute.

Thin slab molds are made of four different plates: two wide plates and two lateral plates, all made of copper (see Figure 1.1.3). In general, lateral plates can be moved or changed to modify the slab section dimensions. The geometry of these plates is more complex than one can expect: they have drilled channels where the cooling water flows, slots in the outside face for thermal expansion, thermocouples, and fastening bolts. To compensate the shrinkage of the slab with the cooling and minimize the gap, the molds are tapered. Moreover, the upper portion of the mold forms a funnel to accommodate the SEN.

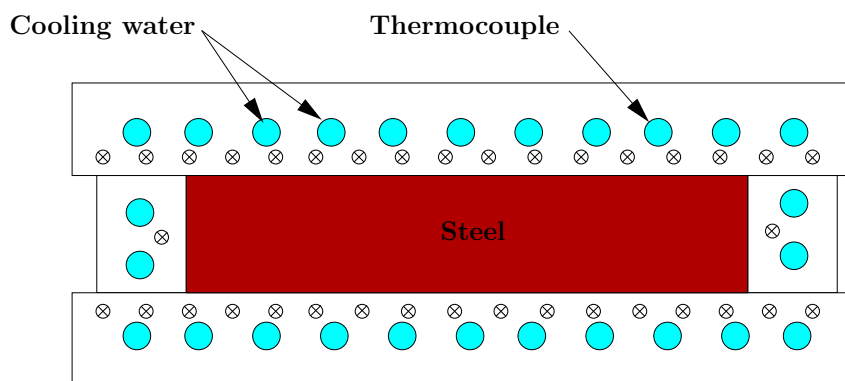


Figure 1.1.3: Schematic of a horizontal section of the mold. We denote in red the inner region of the mold where the steel is tapped. From this schematic, we also see the drilled channels where the cooling water flows and the position of the thermocouples in the mold plates.

In the rest of this chapter, we first discuss the problem of controlling the mold behaviour for a safe and productive functioning of a continuous caster. Then, we discuss the physical phenomena happening in the mold region and the state of the art in their modeling. Later, we consider the mold monitoring problem in an inverse problem and data assimilation setting. To conclude, we summarize the objectives and contributions of this investigation, and present the thesis outline.

1.2 CONTROL OF THE MOLD

As shown in Figure 1.1.2, several phenomena related to steel flow, solidification, stresses and heat transfer appear in the mold region. This complexity makes the mold the most critical part of the CC process. Thus, when running a continuous caster, productivity and safety issues must be addressed at the mold.

Regarding quality, the presence of imperfections on the external surface of the casted piece (cracks, inclusions, etc.) must be avoided. In fact, since casted pieces are generally laminated in later productions, surface defects would become evident affecting also the mechanical properties of the final products.

Due to the creation of the solid skin, a frequent problem arising during CC is the sticking of the steel to the mold. After the detection of a sticking, the casting speed is reduced to reestablish the desired metal flow before restoring the nominal casting speed. This affects the product quality and the productivity of the caster. Also, if not detected on time, it can lead to dangerous events forcing the shutdown of the caster.

Less frequent but more catastrophic events are the liquid break-out and the excessive increase of the mold temperature. The former is due to a non-uniform cooling of the metal. Thus, the skin in some regions of the slab is so thin that it breaks. The latter is generally considered as the most dangerous event in a casting plant. In fact, if the mold temperature is high enough to cause the boiling of the cooling water, we have a dramatic decrease in the heat extraction. Then, the temperature in the mold would quickly rise causing the melting of the mold itself, eventually. Both these incidents are very dangerous and costly. In fact, they generally require the shutdown of the caster, the substitution of expensive components and an extended turnaround.

For all these reasons, the early detection of problems in the mold is crucial for a safe and productive operation of continuous casters. Moreover, their early detection becomes more difficult and relevant as casting speed (thus productivity) of the casters increases.

Since, continuous casters have been running for decades, CC operators already faced all these problems. To have insight of the scenario in the mold, they provided the molds with a measuring equipment. In particular, they measure the pointwise temperature of the mold by thermocouples (see Figure 1.1.3) and the cooling water temperature as well as its flow at the inlet and outlet of the cooling system. Finally, they also measure the height of the meniscus at the top of the mold (see Figure 1.1.2) to know how full the mold is.

The way CC operators use the data coming from the measurement equipment is the following. The thermocouples' temperatures are used to have insight of the mold temperature field. On the other hand, the water temperature rise is used to approximate the heat extracted from the steel by assuming the process to be at steady-state and making an energy balance (we will explain it in detail in Section 3.2).

This approach allowed operators all over the world to run continuous casters for decades. Nevertheless, it has several drawbacks: it relies on the experience of operators, gives very limited information about the heat flux at the mold-slab interface,

and is customized for each geometry so it requires new effort to be applied to new designs. So, a new tool for analyzing the mold behavior is necessary.

In presenting the goal of this investigation, we begin by reporting that CC operators consider that knowing the local heat flux between mold and slab is the most important information to monitor the mold behaviour. In fact, it allows the operator to detect any unwanted behaviour of the steel within the mold. This quantity cannot be experimentally measured since it is not possible to make direct measurements in the solidification region. The only measurements available are temperature measurements by thermocouples that are buried inside the mold plates and cooling water temperature measurements. Thus, our goal is to develop a mathematical tool that, given these measurements, estimates the sought heat flux.

By considering the mold itself to be our computational domain and focusing our interest on its thermal behavior, the mold-slab heat flux can be seen as the Neumann Boundary Condition (BC) on a portion of the boundary. To compute its value, we pose the following inverse problem: given the temperature measurements provided by the thermocouples, estimate in real-time the boundary heat flux at the mold-slab interface that minimize a distance between the measured and simulated temperature at the thermocouples points.

To tackle this complex problem, we can think of dividing it into three related ones:

- Accurate modeling of the thermal problem in the physical mold.
- Solution of the inverse problem of estimating the heat flux.
- Reduction of the computational cost of the inverse problem solution to achieve real-time computation.

In the rest of this chapter, we discuss more in details the above problems giving an overview on their state of the art.

1.3 PHYSICAL PROBLEM

In this section, we give a description of the physical phenomena that occur in the mold region of a caster. Then, we give an overview on previous efforts in modeling them and, finally, we present the physical problems that we will consider in the present work.

Going from the inside to the outside of the mold, we encounter several physical phenomena (see Figure 1.1.2). In the inner part of the mold, we have the liquid pool of steel. There, we have a molten metal flow with dispersed argon bubbles and

inclusion particles. All around the liquid pool, we encounter the solid skin and, in between, the mushy region. Here, the steel changes phase undergoing solidification. Between the steel and the mold, there is a thin layer of flux powder which is liquid close to the steel and solid where in contact with the mold. Finally, we encounter the mold which is surrounding the flux powder (in case of not perfect casting, we can also have an air gap between the mold and the slab).

The mold is composed of a solid (copper) region and a liquid region (water) representing its cooling system. In the copper, we have heat conduction due to the temperature gradients. In the water, we have an incompressible flow inside tubes. To prevent the water from boiling, it is pumped at a very high pressure and flow rate. Therefore, a turbulent flow with high Nusselt number is expected.

According to the previous description, the main physical phenomena for CC include (see [1]):

- Fully-turbulent, transient fluid motion in a complex geometry (SEN, strand liquid pool), affected by dispersed particles and thermal buoyancy.
- Thermodynamic reactions.
- Multiphase flow and heat transport.
- Dynamic motion of free liquid surfaces and interfaces.
- Thermal, fluid and mechanical interactions between solids, liquids and gases.
- Heat transfer in solids.
- Distortion and wear of the mold.
- Solidification of steel shell.
- Shrinkage of the solidifying steel shell.
- Crack formation.

1.3.1 State of the Art

Due to its complexity, the literature on CC mold modeling is extensive. For each physical phenomenon in the previous list, we have at least a dedicated model and several investigations. For this reason, a number of review articles describing the

models used for CC appeared (see e.g., [153, 154]), and we therefore include only a small selection of particularly relevant works.

In particular, Meng and Thomas [93] investigated the modeling of transient slag-layer phenomena in the steel-mold gap. They coupled a heat transfer model in the mold, gap, and steel with analytical transient models of liquid slag flow and solid slag stress to predict transient shear stress, friction, slip, and fracture of the slag layers. Thomas et al. [156] studied the steel flow and solidification in the mold including the transport of superheat with the turbulent transient flow of molten steel, surface level fluctuations, and the transport and entrapment of inclusion particles. Huespe et al. [70] developed a 2D model to simulate the CC of round billets with the objective of determining stresses and strains inside the solidified material at the early stage of casting. Their thermal analysis took into account phase-change in the material and heat transfer through the mold. For the mechanics, they used two models assuming elastoplastic and viscoplastic hardening materials.

A number of works is dedicated to the modeling of the molten steel fluid flow and solidification. The modeling of the molten steel flow is the subject of the review paper of Yuan [170]. In general, the steel is assumed an incompressible Newtonian fluid and the most popular approach is to solve steady-state, single-phase, Reynolds-Averaged Navier-Stokes (RANS) equations together with a turbulence model (see [112]). However, Large Eddy Simulation (LES) simulations have also been performed especially for the transient case when RANS becomes less accurate (see [30]).

The solidification of the steel is obviously related to the molten steel flow in the inner region. Then, to model the solidification is a complex, coupled problem and, as such, has been intensively investigated in the literature. In general, this is a phase change problem, so a vast literature is available on the subject (see e.g., [6, 74, 81, 82, 85, 88, 95, 100, 147, 151, 169, 175–177]). A detailed description of the solidification in casting is available in Stefanescu's monograph [146], while a review on the initial solidification models was done by Wang [161].

Since steel is not a pure metal, its solidification is not isothermal but is characterized by the presence of a mushy region. Moreover, the steel solidification is affected by several factors, mainly: the mold heat extraction, the slag layer, the molten steel temperature and velocity field. All this makes modeling the steel solidification a difficult task.

In general, the slab can be considered as a solid-liquid two-phase system. In order for this system to be fully described, equations expressing mass, momentum, solute and energy transport have to be developed for each phase. A general two-phase description of a binary solidification system was developed in the well-known two-

fluid model (see e.g., Carver [24]). In this model, separate governing equations for each phase (solid and liquid) are developed. Then, they are coupled through inter-phase transfer terms.

From the above model, Voller [160] developed a set of one-phase models for steel solidification. These models were derived making the following assumptions: (i) Newtonian and laminar flow, (ii) homogeneous and isotropic properties in the phases, (iii) effects of small disturbances in the fields (e.g., dispersion fluxes, supercooling of the liquid, etc.) are neglected, and (iv) thermodynamic equilibrium at the solid-liquid interface. In these models, the variables are usually the “mixture” properties. The models developed by Voller are: the mushy fluid model, the columnar dendritic with dispersed microstructure and the columnar dendritic with distinct microstructure. The former model is often used in modeling solidification in CC molds (see e.g., [83]).

Voller’s models assume a laminar flow, however solidification of turbulent flows (common in CC) was also investigated. Prescott and Incropera [121, 122] developed a low Reynolds number $k - \epsilon$ model to handle turbulence during solidification. They used a simple approach modifying the turbulence kinetic energy in the domain which may include a mushy zone. Within a coherent mushy zone, turbulence is assumed dampened by shear which is linearly correlated with the reduction of the mush permeability. The influence of turbulence on the momentum and energy transports is then considered by an effective viscosity and thermal conductivity. Using this model, Wu et al. [166] tested on a 2D benchmark case the effect of turbulence on the thickness of the solid skin and mushy region. They concluded that the turbulence plays an important role near the front of the mushy zone and in the downstream region leading to a significantly smaller mushy region.

This was just an overview on well established solidification models in CC. More sophisticated models have been developed for alloy solidification in which complex, thermal-fluid-structure coupling was considered. We refer the interested reader to [8].

After an overview on modeling phenomena inside the mold, we now discuss the modeling of the mold itself. The most popular model was developed by Samarasekera and Brimacombe [133]. They developed a model for billet molds considering a slice of the mold in the casting direction, so the domain is 2D. The mold is assumed in steady-state, the water in plug flow, and thermal properties of the materials are assumed independent of temperature. Then, the model is a 2D heat conduction model in the solid coupled by a Robin BC to a sectional heat balance for the water.

Du et al [37–39], studied the thermal and mechanical behaviour of a slab mold. For the thermal model, they assumed that the heat flow along the casting direction is negligible, whereas it varies along the horizontal direction. The top and bottom

of the mold were considered to be adiabatic and the material thermal properties constant. Thus, they used a 2D transient heat conduction model. For the stress model, they assumed that the mold is in plane stress condition, the mechanical properties are temperature dependent, and there is no creeping. They used the thermal model to compute the temperature field. Then, the obtained temperature field was loaded into the stress model for stress and distortion analysis. Similarly, Park et al. [114] used a 3D transient heat conduction model for the mold copper plate with convective BC for the water cooling and assuming linear increase in the water temperature. The computed temperature field was then used to compute the thermal effects on the deformation of the mold.

Gupta et al. [63] studied the effect of mold coating and geometry on the thermal field by using a steady-state 3D heat conduction model. As BCs, they applied an empirical mold-slab heat flux formula, convective BC for the cooling water assuming linear temperature increase in the water, and adiabatic BC on the rest of the boundary.

Xie et al. [168] studied the heat transfer in continuous slab molds by also modeling the water flow. They ignored the taper and the bolt holes of the mold, and assumed that all the heat is extracted by the water. As usual, radiate heat was ignored. The water flow was assumed steady and incompressible, and the slots in which the water flows are assumed straight lines. Then, they coupled a heat conduction problem in the solid coupled with an advection-diffusion model for the temperature in the water by a Robin transmission condition. The velocity in the water was computed by RANS equations with $k - \epsilon$ turbulence model.

1.3.2 Specific Physical Problem

As discussed in Section 1.2, our goal is to monitor the real-time behavior of CC molds by estimating the mold-slab heat flux. To tackle this task, we require a mold model. We could think of modelling all the phenomena discussed in Section 1.3 from the SEN to the secondary cooling region. However, the resulting model would be quite complex and computationally expensive to deal with, especially for real-time applications. Then, our approach is to have the mold plates as computational domain and the unknown heat flux as a BC.

In the rest of this section, we consider the problem of developing a physical model for CC molds. Firstly, we present a set of general assumptions. Then, we propose a hierarchy of models by adding further assumptions to the previous set.

In modeling the thermal behavior of the mold, we consider the following well established assumptions (see [133, 154]):

- The copper mold is assumed a homogeneous and isotropic solid material.
- The cooling water temperature is known.
- The thermal expansion of the mold and its mechanical distortion are negligible.
- The material properties are assumed constant.
- The boundaries in contact with air are assumed adiabatic.
- No boiling in the water is assumed.
- The heat transmitted by radiation is neglected.

The adiabaticity of the boundaries in contact with air is justified when considering the magnitude of the heat extracted by the cooling water when compared to the one extracted by the still air around the mold. A similar justification, can be used for neglecting the mold radiation. Considering the thermal conductivity constant comes also from a practical consideration. CC molds are generally made by copper and they work in a temperature range in between 600 K and 800 K . In this range the thermal conductivity varies of about 2%. Thus, this is the maximum error coming from this assumption.

Finally, since we want to have solution in real-time (e.g., at each second) and the casting speed is of few meters per minute, we consider steady-state models at first. Then, we will relax this assumption. Moreover, we only consider 3D mold models because we are interested in the heat flux in all the mold-slab interface.

As a final remark, the running parameters of the cooling system and its geometry ensure a fully developed turbulent flow. In fact, these molds are equipped with a closed loop cooling system where the water is pumped at a high pressure. The average velocity in each cooling channel is approximately 10 m/s , the diameter being approx. 10 mm . Thus, the Reynolds number in the cooling system is around 10^5 , which ensures a turbulent flow.

According to the mentioned assumptions, we consider the following physical problem:

- (MS1) The domain is composed of the (solid) copper mold and (liquid) cooling water. Then, a liquid-solid steady-state three-dimensional heat transfer model one-way coupled with a turbulent fluid flow model for the fluid dynamics of the cooling water is considered.

The one-way coupling is in the sense that we neglect the effects that changes in the fluid temperature have on the fluid dynamics (i.e. buoyancy).

Notice that in the mold, we consider the cooling water flowing in circular channels with constant cross section. Then, we can consider simplified 1D models for the turbulent flow in tubes (see e.g., [164], Chapter 7). These models give the shape of the averaged boundary layer in a tube and the velocity has only one component, the vertical component along the axis of the tube. Thus, we must pay attention in making this simplification for the following model since we neglect the heat transport due to turbulence eddies and mixing. Therefore, we derive the following simplified version of the previous model:

- (MS2) The domain is composed of the (solid) copper mold and (liquid) cooling water. Then, a liquid-solid steady-state three-dimensional heat transfer model one-way coupled with a simplified tube flow model for the fluid dynamics of the cooling water is considered.

Thanks to the high Reynolds number of the flow, we can further assume that the cooler and hotter water molecules are well mixed. Consequently, the temperature in each section of the cooling channel is approximately constant. Moreover the water is pumped in a closed circuit, the water flow rate being constant. In turn, since the channels have constant section, the velocity of the fluid is also uniform and constant (plug flow). Making these assumptions, we end up with the following model which is a three dimensional version of the 2D model proposed by Samarasekera and Brimacombe [133]:

- (MS3) The domain is composed of the (solid) copper mold and (liquid) cooling water. A steady-state three-dimensional heat conduction model with a convective BC on the portion of the boundary in contact with the cooling water is considered. Assuming constant temperature in each transversal section of the cooling system, the water thermal model is given by a cross sectional heat balance. In this case the two models are two-way coupled by the boundary condition.

Finally, considering that the temperature increase of the cooling water is of only few degrees, we can assume that the water temperature is known. Then, we consider the following simplification of the previous model:

- (MS4) The computational domain is only composed of the (solid) copper mold. We consider a steady-state three-dimensional heat conduction model with a convective BC in the portion of the boundary in contact with the cooling water. The water temperature is known at the inlet and outlet of the cooling system. The water temperature is assumed to be linear.

As previously mentioned, we now drop the steadiness assumption to develop an unsteady-state version of model (MS4):

- (MU4) The computational domain is only composed of the (solid) copper mold. We consider an unsteady-state three-dimensional heat conduction model with a convective BC in the portion of the boundary in contact with the cooling water. The water temperature is known at the inlet and outlet of the cooling system. The water temperature is assumed to be linear in space and constant in time.

We have now available a hierarchy of models which we built by adding assumptions. Figure 1.3.1 illustrates this models hierarchy.

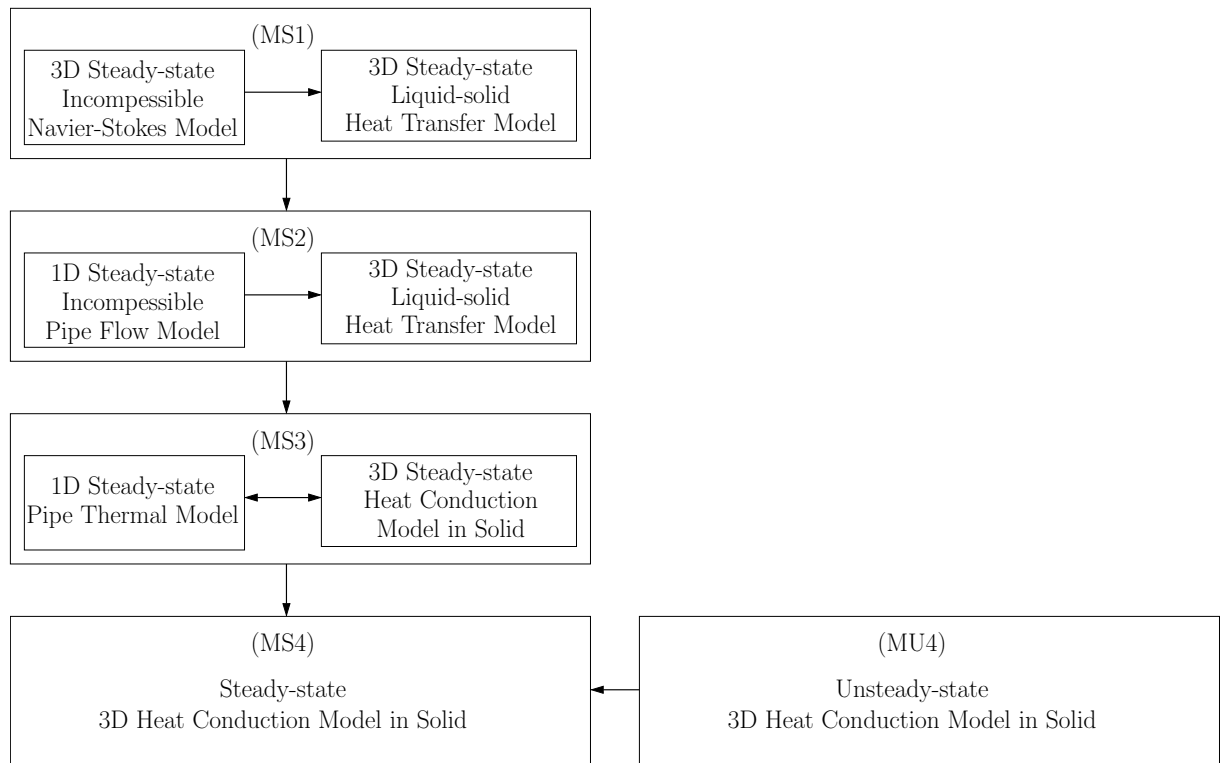


Figure 1.3.1: Scheme of the models hierarchy for the CC mold.

1.4 ESTIMATING THE MOLD-SLAB HEAT FLUX

Having defined the physical problem, we consider that the main objective of the present research is to compute in real time the heat flux at the steel-mold interface from the temperature monitoring data in the industrial plant. The thin slab casting molds that we consider are equipped with a number of thermocouples. These thermocouples are all buried at the same distance (few millimeters) from the hot face of

the mold and they measure the temperature at a frequency of 1 Hz. As usual in experimental measurements, the measured temperature is affected by errors that depend on the thermocouple type, accuracy and tolerance range.

The objective of our research is to use these measured temperatures to estimate the boundary heat flux. From a mathematical point of view, problems of this kind are generally called Inverse Heat Transfer Problem (IHTP). A characteristic of these inverse problems is that their solution could not exist, could be not unique and could not depend continuously on the data. Thus, they are not well-posed in the sense of Hadamard (see [99]) but instead they are ill-posed problem.

The ill-posedness of these inverse problems requires a careful handling and the application of regularization techniques to recover well-posedness. In this section, we review previous efforts in solving similar inverse problems with a focus on IHTPs.

1.4.1 State of the Art

The literature on IHTPs is vast. For a detailed review we refer to [2, 11, 26, 111]. As already mentioned, also the problem of computing the mold-slab heat flux from temperature measurements in the mold is not new in the literature (see e.g., [90, 127, 158]). From a mathematical point of view, this problem is the estimation of a Neumann BC (the heat flux) having as data pointwise measurements of the state variable in the interior of the domain. Such problems were also addressed in investigations not related to CC (see e.g., [129, 159]). Due to the vastness of the literature on the subject, we focus on previous efforts in BC estimation in IHTPs and mold-slab heat flux estimation, giving only few references about the general inverse problems framework.

Research in IHTPs started in the 50s. It was driven by the interest in knowing thermal properties of heat shields and heat fluxes on the surface of space vehicles during re-entry. From a heuristic approach in the 50s, researchers moved to a more mathematically formal approach. In fact, in the 60s and 70s, most of the regularization theory that we use nowadays to treat ill-posed problems was developed (see [2, 9, 10, 31, 157]). Here, we discuss in general the most popular methodologies used for the solution of IHTPs.

A possible classification of inverse problems is by the kind of object to be estimated. In particular, Dantas and Orlando [36] divided them into four categories: boundary problems, regarding the estimation of BCs; coefficient problems, concerning with the estimation of physical properties; retrospective problem, estimating the initial conditions; and geometric problems, dealing with the estimation of geometric

characteristics of a body. In general, two approach can be used for their solution: (i) to use a deterministic least square approach, stating the problem as the minimization of a functional and using the optimal control framework, or (ii) to use a statistical approach, stating the problem in the Bayesian framework.

Using a deterministic approach, we want to estimate a function such that it minimizes a measure of the distance between the state variable (e.g., temperature) that we measure and the one that we compute. This is the most used approach because we can state the inverse problem in the optimal control framework. In this setting, we can exploit for the solution of the problem several techniques that were proposed in the literature. Namely, the most popular methods are Conjugate Gradient Method (CGM) [101], Levenberg-Marquardt Method (LMM) [123] and Tikhonov regularization [61]. The choice of the methodology to use depends, in general, from the type of inverse problem and the object to estimate.

In the Bayesian setting, we treat the object to estimate as a random variable. Then, by the Bayes theorem, its conditional probability with respect to the data we measure (posterior) is given by the prior probability times the likelihood. The prior probability contains the knowledge we have of the object to estimate (e.g., range of values), while the likelihood contains the direct problem. With this approach, we put all the information we have (e.g., the physical model, the measurements, the statistical model of the noise) inside the prior probability and the likelihood, obtaining the probability of the quantity we want to estimate. This approach is very useful to treat uncertainty in the data and/or in the model. However in general, it requires some very expensive Monte Carlo methods to sample the posterior probability which are well known for being computational expensive.

To provide an overview of the previous efforts in solving the present inverse problem, we briefly describe some works related to the estimation of boundary heat flux in CC molds. Traditionally (see [90]), a heat flux profile was selected and by trial and error adapted to match the measured temperatures. Pinhero et al. [119] were the first to use an optimal control framework and regularization methods. They used a steady-state version of the 2D mold model proposed by Samarasekera and Brimacombe [133] and parameterized the heat flux with a piecewise constant function. Finally, they used Tikhonov's regularization for solving the inverse problem and validated the results with experimental measurements. A similar approach was used in [94]. Ranut et al. [127, 128] estimated the heat flux transferred from the solidifying steel to the mold wall both in a 2D and 3D domain. They used a steady-state heat conduction model for the mold and parameterized the heat flux with a piecewise linear profile in 2D and symmetric cosine profile in 3D. For the solution of the inverse

problem, they used the CGM and a mixed GA-SIMPLEX algorithm (see [104]) in 2D while in 3D they only used the GA-SIMPLEX algorithm. Their results were also tested with experimental data.

Hebi et al [67, 91] attempted to estimate the solidification in round billet CC by using a 3D transient heat conduction model in the strand and the mold with a Robin condition at the mold-strand interface. Then, they planted the following inverse problem: find the Heat Transfer Coefficient (HTC) between mold and strand such that a distance between measured and computed temperatures at the thermocouples is minimal. They assumed the HTC to be piecewise constant. It was firstly set to a uniform value. Then, by using sensitivity coefficients, each piece was iteratively adapted to match the measured temperature. To allow convergence, a relaxation factor was introduced in between the iterations. They validated the results with plant measurements without obtaining a good agreement. A similar approach was used by Gonzalez et al. [62] and Wang et al. [69, 152, 162, 174], the latter using a Neumann condition at the mold-strand interface.

Wang and Yao [163] used the aforementioned inverse problem solution technique to estimate the HTC in round billet CC mold. Then, they used the results obtained to train a Neural Network (NN) for on-line computation. Similarly, Chen et al [32], used the fuzzy inference method for estimating the mold heat flux. They modeled the mold with a 2D steady-state heat conduction model in the solid and parameterized the boundary heat flux. They tested the results on a numerical benchmark obtaining a good agreement.

Yu and Luo [76] considered a 2D vertical section of a slab and the solidification problem therein. They developed a modified Levenberg–Marquardt method to estimate the HTC in the secondary cooling region from temperature measurements on the surface of the slab.

Udayraj et al. [158] applied CGM with adjoint problem for the solution of the inverse steady-state 2D heat conduction problem, this method was first proposed by Alifanov [2] for the regularization of boundary inverse heat transfer problems. By using this method there is no need of parameterizing the heat flux. However, the method underestimate the heat flux away from the measurements. To overcome this issue, the authors proposed to average the computed heat flux at each step and use the uniform averaged value as initial estimation for the following step. Similarly, Chen et al. [33] tackled the problem of estimating the steady boundary heat flux in 2D circular CC supporting rollers based on temperature measurements inside the domain. For its solution, they used the CGM proposed by Alifanov [2].

We conclude this section by describing previous works that are related to the

present research but not to CC. Ambrosi et al. [3, 159] studied the mathematical formulation of the force traction microscopy problem. This inverse problem consists in obtaining the boundary stress field on a portion of the boundary (Neumann BC) based on the pointwise measurement of the displacement (state variable) inside the domain. The similarity with the present research is the presence of pointwise observations and a boundary inverse problem with (linear) elliptic direct problem. In [3], they stated the 2D direct problem and the related inverse problem in the standard optimal control framework due to Lions [87] for which the unknown BC is the distributed boundary control. Then in [159], they extended the formulation to the 3D linear elasticity model proving existence and uniqueness of the optimal control solution.

1.5 OBJECTIVES

From the previous overview on the state of the art, we appreciate that the problem of estimating the mold-steel heat flux as been addressed by some researchers already. However, we could not find in the literature a reliable solution to this problem that is able to reconstruct the heat flux in all the mold-steel interface simultaneously and to provide the solution in real-time. To solve this industrial problem, the goal of the present work is to develop mathematical tools able to estimate the mold-steel heat flux in real-time given the thermocouples measurements.

To reach this goal, we split it in four interconnected objectives: the accurate modelling of the mold, the formulation and solution of the heat flux estimation problem, the reduction of the computational cost to achieve real-time performance, and the test of the developed mathematical tools.

With respect to the mold modelling, we approach this task with the objective of generating a model hierarchy. Starting from a first set of assumptions, we derive the related model for the heat transfer into the mold. Then, we generate a hierarchy of models by adding further assumptions to this initial set. This cutting edge collection of models is by itself a first unprecedented outcome of this thesis.

Once that the mold hierarchy is constructed, we select two models and study the related boundary heat flux estimation problems. In formulating these inverse problems, we propose novel formulations together with the classical one that we could find in the literature. When dealing with their solution, we study the employment of traditional techniques such as Alifanov's regularization together with the development of novel methodologies. In particular, we propose original direct methods that exploit the parameterization of the sought heat flux. The advantage of these methods is twofold. Firstly, they are direct methods that do not require several iterations to find a solution. Secondly, they rely on the offline-online decomposition, meaning

that we have a computational expensive offline phase that can be computed before starting the machinery and a computationally cheap online phase that estimates the mold-slab heat flux.

In some cases, this offline-online decomposition is sufficient to achieve real-time performances. In others, we investigate the development of innovative Model Order reduction (MOR) techniques for the reduction of the computational cost of the online phase to ensure the fulfillment of the real-time constrain.

Also due to the industrial implications of this research, to properly test all the proposed methodologies is crucial. To do it, we design new specific benchmark test cases. These test cases are both academic with analytical solutions and numerical test case in which we mimic the real industrial scenario. We use these benchmarks to perform several different analysis both to validate the simulation of the mold models, and to test and compare the performance of the proposed inverse solvers.

1.6 THESIS OUTLINE

We divide the present thesis in two parts. In the first one, we consider the steady-state mold models and the related inverse problems. In the second, we use the unsteady-state mold model formulating the heat flux estimation problem in this setting.

Each part as a similar structure. Firstly, we introduce the mathematical models to simulate the heat transfer in CC molds for the considered direct problems providing the related literature review. In a following chapter, we formulate the inverse problems for the computation of the boundary heat flux. After a review of the most important results for the mathematical analysis of the proposed inverse problem, we present novel methodologies for the solution of these problems. Moreover, if necessary, we describe the application and development of MOR techniques for the speed-up of the proposed inverse solvers.

Finally, we discuss the design of benchmark test cases. Then, we use them to validate and compare all the different methodologies for the solution of the inverse problems.

Part I

Estimation of the Boundary Heat Flux by Using a Steady-State Mold Model

This part contains partial reproduction of the work
U. E. Morelli, P. Barral, P. Quintela, G. Rozza, and G. Stabile. A numerical approach for heat flux estimation in thin slabs continuous casting molds using data assimilation. *International Journal for Numerical Methods in Engineering*, 122(17):4541–4574, 2021

2 STEADY-STATE DIRECT PROBLEM

In the present chapter, we discuss the mathematical formulation of the steady-state physical models introduced in Section 1.3.2.

We begin this chapter by introducing the domain of interest and the main notation in Section 2.1. In Section 2.2, we present the characteristic physical data of CC molds. Section 2.3 introduces some modeling assumptions which are common to all the mold models of the thesis. In Section 2.4, we introduce the mathematical formulation of model (MS1) and (MS2). These models share the same thermal model, the difference being in the considered fluid flow model. In fact, model (MS1) incorporates an incompressible Navier-Stokes fluid model, and model (MS2) an empirical tube flow model. Both provide the fluid velocity required by the advection-diffusion thermal model. For the involved steady-state thermal model, we prove the existence and uniqueness of the solution.

In Section 2.5, we derive model (MS3), introduced in Section 1.3.2, by adding some assumptions to model (MS1) that allow us to consider a 1D thermal model in the fluid domain. This model was first proposed for the 2D case by Samarasekera and Brimacombe (see [134]); we present here its derivation in the 3D case. Finally, in Section 2.6, we formulate the mathematical description of model (MS4) by adding further simplifying assumptions on model (MS3). The resulting model is a 3D heat conduction model in the solid domain with a convection BC at the portion of the solid boundary in contact with the cooling fluid.

2.1 COMPUTATIONAL DOMAIN AND NOTATION

Consider a fluid and a solid domains, Ω_f and Ω_s , which are open Lipschitz bounded subsets of \mathbb{R}^3 , with smooth boundaries Γ_f and Γ_s and such that $\Omega_f \cap \Omega_s = \emptyset$ (see Figures 2.1.1 and 2.1.2). Let $\Gamma_f = \Gamma_{f_{in}} \cup \Gamma_{f_{out}} \cup \Gamma_{sf}$, where $\overset{\circ}{\Gamma}_{f_{in}}$, $\overset{\circ}{\Gamma}_{f_{out}}$ and $\overset{\circ}{\Gamma}_{sf}$ have mutually empty intersections and $\overset{\circ}{\Gamma}_{f_{in}}$ is non empty. Likewise, let $\Gamma_s = \Gamma_{s_{in}} \cup \Gamma_{s_{ex}} \cup \Gamma_{sf}$ where $\overset{\circ}{\Gamma}_{s_{in}}$, $\overset{\circ}{\Gamma}_{s_{ex}}$ and $\overset{\circ}{\Gamma}_{sf}$ are disjoint sets. Moreover, let $\Gamma_{sf} := \Gamma_f \cap \Gamma_s$ be not empty. Let Ω be defined by

$$\Omega := \overline{(\overline{\Omega_f} \cup \overline{\Omega_s})}, \quad (2.1.1)$$

and $\Gamma = \partial\Omega = \Gamma_{sin} \cup \Gamma_{sex} \cup \Gamma_{fin} \cup \Gamma_{fout}$. The Eulerian Cartesian coordinate vector is denoted by $\mathbf{x} \in \Omega$ and $\mathbf{n}_i(\mathbf{x})$ is the unit normal vector that is directed outwards from Ω_i , $i \in \{f, s\}$.

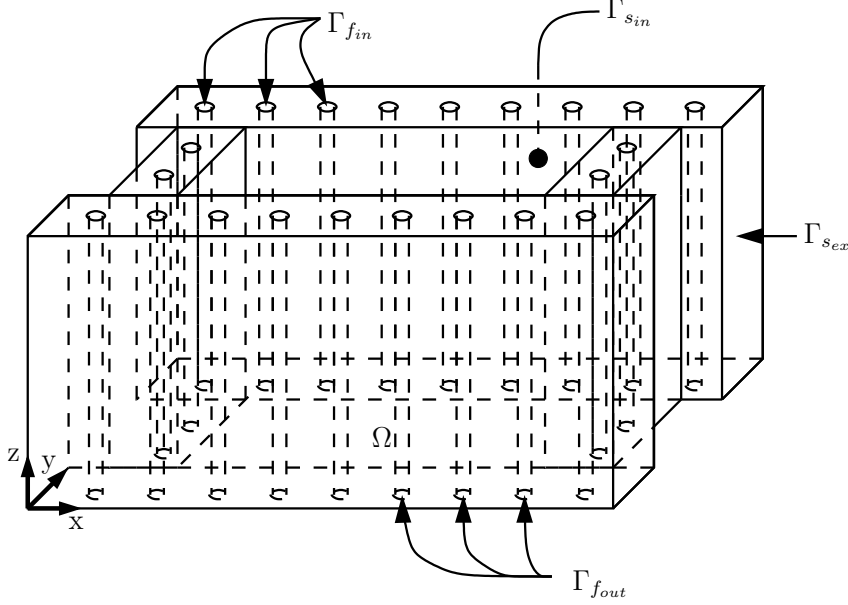


Figure 2.1.1: Schematic of the domain, Ω , and its boundaries. It is divided into two subdomains: the solid domain (mold), Ω_s , and the fluid domain (cooling system), Ω_f .

In this setting, Ω_s and Ω_f correspond to the region of the space occupied by the mold and its cooling system, respectively. The boundary Γ_{fin} is the inlet of the cooling system, while Γ_{fout} the outlet. The interface between the mold and the cooling system is denoted by Γ_{sf} . Finally, Γ_{sin} is the portion of the mold boundary in contact with the solidifying steel.

In the following, we consider the standard Sobolev space $H^1(\Omega)$, with $H^{1/2}(\Gamma)$ being its image given by the trace operator

$$\gamma_\Gamma : H^1(\Omega) \rightarrow H^{1/2}(\Gamma). \quad (2.1.2)$$

Analogously, we consider $H^1(\Omega_i)$, $i \in \{f, s\}$, and $H^{1/2}(\Gamma_{l(i)})$, with $l(f) \in \{fin, fout, sf\}$, and $l(s) \in \{sin, sex, sf\}$.

We now define the space

$$V(\Omega_f) := \{\mathbf{w} \in [H^1(\Omega_f) \cap L^\infty(\Omega_f)]^3 \mid \nabla \cdot \mathbf{w} = 0, \quad \gamma_{\Gamma_{sf}} \mathbf{w} \cdot \mathbf{n}_f = 0\}, \quad (2.1.3)$$

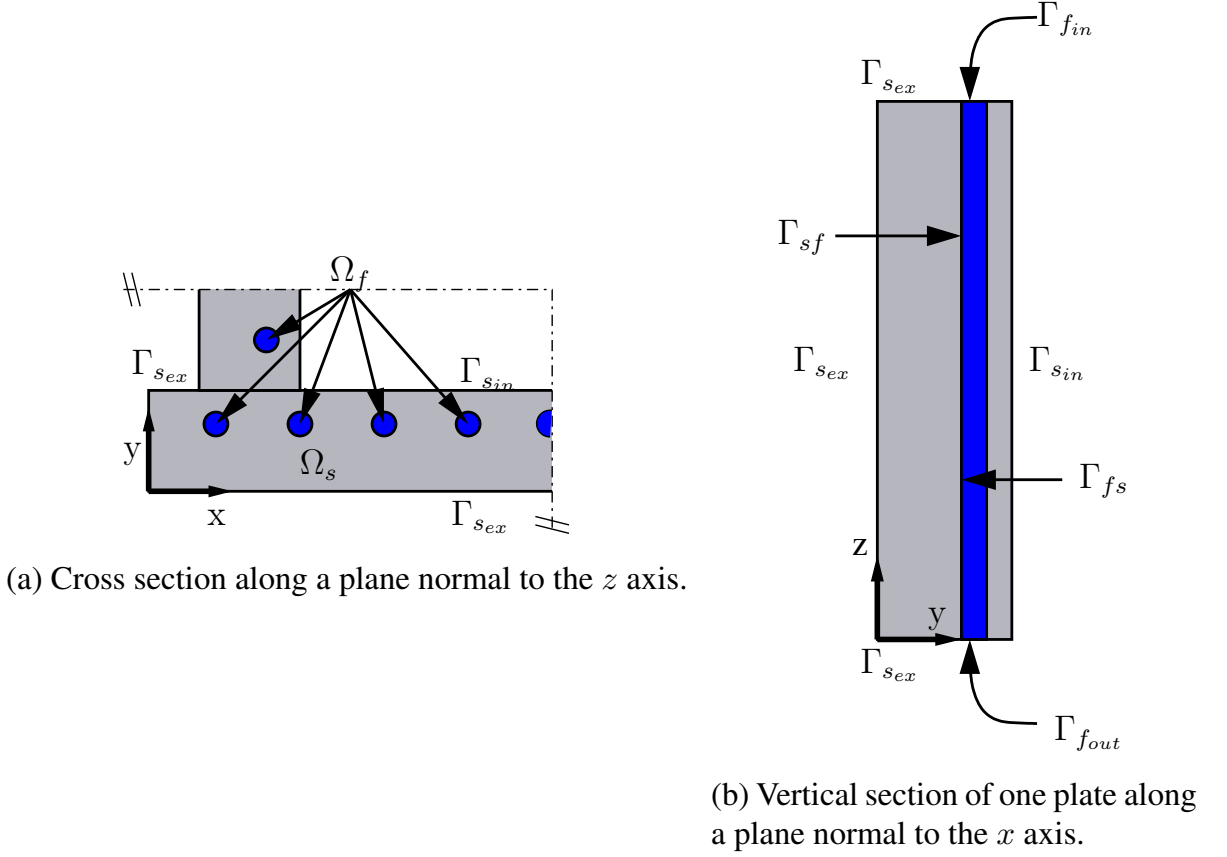


Figure 2.1.2: Cross sections of the domain, Ω . For clarity, the fluid domain is colored in blue and the solid domain in gray.

Moreover, we define the Hilbert spaces (see [17])

$$\begin{aligned}
 H_{\Gamma_{f_{in}}}^1(\Omega) &:= \{w \in H^1(\Omega) \mid \gamma_{\Gamma_{f_{in}}} w = 0\}, \\
 \mathcal{H}^1(\Omega) &:= H^1(\Omega_s) \times H^1(\Omega_f), \\
 \mathcal{H}_{\Gamma_{f_{in}}}^1(\Omega) &:= H^1(\Omega_s) \times H_{\Gamma_{f_{in}}}^1(\Omega_f),
 \end{aligned} \tag{2.1.4}$$

with the standard norm

$$\|(w_s, w_f)\|_{\mathcal{H}^1(\Omega)} = (\|w_s\|_{H^1(\Omega_s)}^2 + \|w_f\|_{H^1(\Omega_f)}^2)^{1/2}. \tag{2.1.5}$$

We identify $H^1(\Omega)$ (resp. $H_{\Gamma_{f_{in}}}^1(\Omega)$) with the subspace of $\mathcal{H}^1(\Omega)$ (resp. $\mathcal{H}_{\Gamma_{f_{in}}}^1(\Omega)$) of the functions with equal trace at the interface Γ_{sf} (see [17]). So that, a function z in $H^1(\Omega)$ can be identified by the pair (z_s, z_f) of its restrictions to the domains Ω_s and Ω_f . Taking into account that $\Gamma_{f_{in}} \subset \Gamma$ and since the trace operator from

$H^1(\Omega)$ onto $H^{1/2}(\Gamma_{f_{in}})$ is continuous and surjective, there exists a lifting operator. Therefore, for a \tilde{w} in $H^{1/2}(\Gamma_{in})$ there exists $R_{\tilde{w}}$ in $H^1(\Omega)$ such that (see [124])

$$R_{\tilde{w}} = (R_{\tilde{w}_s}, R_{\tilde{w}_f}) \in H^1(\Omega), \quad \gamma_{\Gamma_{f_{in}}} R_{\tilde{w}} = \tilde{w}. \quad (2.1.6)$$

Moreover, there exists a constant $C > 0$, such that

$$\|R_{\tilde{w}}\|_{H^1(\Omega)} \leq C \|\tilde{w}\|_{H^{1/2}(\Gamma_{f_{in}})}, \quad \forall \tilde{w} \in H^{1/2}(\Gamma_{f_{in}}). \quad (2.1.7)$$

The objective of the direct problem is to compute the absolute temperature in Ω . In what follows, we denote the absolute temperature by T . Consequently, the absolute fluid and solid temperatures are denoted by T_f and T_s respectively. Moreover, we denote the fluid velocity by \mathbf{u} and the fluid pressure by p . Finally, let us define the fluid deformation rate tensor by

$$D[\mathbf{u}] := \frac{\nabla \mathbf{u} + \nabla \mathbf{u}^T}{2}. \quad (2.1.8)$$

2.2 PHYSICAL DATA

In this section, we provide some physical data characteristic of CC molds. We use them in the next sections to motivate some modeling assumptions and simplifications. Table 2.1 summarizes the typical values of different physical quantities for a CC mold. The presented copper properties are for a metal temperature of 600 K (see [47]), while the water properties are for a temperature of 313 K and a pressure of 10 bar . These are the nominal operating temperatures in a caster.

We define the following quantities:

- The average cross section velocity for a fluid flowing through a tube is given by

$$u_{avg} := \frac{Q}{A_{tube}}, \quad (2.2.1)$$

where Q is the cooling water flow rate and A_{tube} its cross sectional area. Note that since the flow rate and the area are constant, u_{avg} is also constant.

- The Reynolds number in the cooling tubes is

$$Re := \frac{2\rho_f u_{avg} d_{tube}}{\mu_f}, \quad (2.2.2)$$

where d_{tube} is the tube diameter. Considering the values in Table 2.1, in the cooling tubes it is $1.52e5$. Since it is above $1e4$, we have a fully developed turbulent flow (see [25]).

- Finally, another important dimensionless quantity is the Peclet number. We define it as

$$Pe := \frac{\rho_f C_{p_f} d_{tube} u_{avg}}{k_f}. \quad (2.2.3)$$

According to Table 2.1, in the cooling tubes we have $Pe = 6.6e5$.

Table 2.1: Typical physical parameters for CC molds.

Physical domain	Parameter	Value
Solid (copper)	Thermal conductivity [47], k_s	383 W/(mK)
	Density [44], ρ_s	8940 kg/m ³
	Specific heat capacity [42], C_{p_s}	390 J/(kgK)
Fluid (water)	Thermal conductivity [48], k_f	0.63 W/(mK)
	Density [43], ρ_f	992 kg/m ³
	Specific heat capacity [46], C_{p_f}	4180 J/(kgK)
	Dynamic viscosity [45], μ_f	6.54e-4 Pa s
	Design inlet temperature, \tilde{T}_f	313 K
	Design flow rate per tube, Q	1 l/s
	Tubes diameter, d_{tube}	0.01 m
	Area of tubes cross section, A_{tube}	7.85e-5 m ²
	Average cross section velocity, u_{avg}	10 m/s
	Reynolds number, Re	1.52e5
Peclet number, Pe	6.6e5	

2.3 GENERAL ASSUMPTIONS FOR CC MOLD MODELS

In this section, we introduce some assumptions that are common to all the models of this chapter.

We shall assume all along the following assumptions on the data:

- (H1.1) The thermal conductivity is constant and strictly positive in each sub-domain: $k_f \in \mathbb{R}^+$ and $k_s \in \mathbb{R}^+$.
- (H1.2) The fluid density and its specific heat are constant and strictly positive: $\rho_f \in \mathbb{R}^+$, $C_{p_f} \in \mathbb{R}^+$.
- (H1.3) The solid density and its specific heat are constant and strictly positive: $\rho_s \in \mathbb{R}^+$, $C_{p_s} \in \mathbb{R}^+$.
- (H1.4) The mold-steel heat flux, g , belongs to $L^2(\Gamma_{sin})$.

- (H1.5) The fluid temperature at the inlet boundary, \tilde{T}_f , is known and belongs to $H^{1/2}(\Gamma_{f_{in}})$.
- (H1.6) The heat source in the fluid, f_f , and in the solid, f_s , belong to $L^2(\Omega_f)$ and $L^2(\Omega_s)$, respectively.
- (H1.7) The dynamic viscosity of the fluid is constant and strictly positive: $\mu_f \in \mathbb{R}^+$.
- (H1.8) The fluid flow is assumed incompressible, Newtonian and at steady-state.

In what follows, we present a hierarchy of CC mold models. This hierarchy is created adding assumptions to (H1.1)-(H1.8).

2.4 THREE-DIMENSIONAL STEADY-STATE LIQUID-SOLID HEAT TRANSFER MODEL

In this section, we consider the physical problems (MS1) and (MS2) introduced in Section 1.3.2. As already mentioned, they share the same thermal model, the difference being in the considered fluid model. We recall that the thermal and the fluid models are assumed to be one-way coupled. Then, we discuss the formulation of the fluid flow and thermal models that arise in CC molds.

The CC mold heat transfer problem is characterized by a forced heat convection region (the cooling water) in contact with a heat conduction region (the solid part of the mold). This physical problem is commonly referred in the literature as a conjugate heat transfer problem (see [89]).

Since the thermal and the fluid models are one-way coupled, we firstly need to compute the fluid velocity. Then, we input it into the thermal model. As described in the previous chapter, we consider the steady-state incompressible Navier-Stokes equations for modeling the fluid flow in model (MS1) in Section 2.4.1. While, for model (MS2), we replace it by a simplified tube model in Section 2.4.2. Both these fluid models are well established in the literature (see e.g. [164]). Finally, we discuss in Section 2.4.3 the mathematical formulation of their common thermal model with the corresponding proof of solution uniqueness.

2.4.1 Incompressible Navier-Stokes Flow Model

According to the assumptions made, the fluid must satisfy the steady-state Navier-Stokes equations for an incompressible flow. Being a viscous flow, we impose non slip BC at the solid wall, Γ_{sf} . Moreover, we assume that the inlet velocity and pressure are known on $\Gamma_{f_{in}}$ and that the flow is fully developed at the outlet $\Gamma_{f_{out}}$ (see [79]). Then, we consider the following

Problem 2.1. Find \mathbf{u} and p defined in Ω_f such that

$$\begin{aligned} \nabla \cdot \mathbf{u}(\mathbf{x}) &= 0, & \text{in } \Omega_f, \\ \rho_f \nabla \cdot [\mathbf{u}(\mathbf{x}) \otimes \mathbf{u}(\mathbf{x})] &= -\nabla p(\mathbf{x}) + \nabla \cdot (2\mu_f(\mathbf{x})D[\mathbf{u}](\mathbf{x})) + \rho_f \mathbf{b}(\mathbf{x}), & \text{in } \Omega_f, \end{aligned} \quad (2.4.1)$$

with BCs

$$\begin{cases} \mathbf{u}(\mathbf{x}) = \mathbf{u}_{in}(\mathbf{x}) & \text{on } \Gamma_{fin}, & (2.4.2) \\ p(\mathbf{x}) = p_{in}(\mathbf{x}) & \text{on } \Gamma_{fin}, & (2.4.3) \\ \mathbf{u}(\mathbf{x}) = 0 & \text{on } \Gamma_{sf}, & (2.4.4) \\ \nabla \mathbf{u}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = 0 & \text{on } \Gamma_{fout}, & (2.4.5) \end{cases}$$

where $\mathbf{b}(\mathbf{x}) = (0, 0, -9.81) \text{ m/s}^2$ is the gravitational acceleration. This system of equations is known as the "primitive variables" formulation of the incompressible Navier-Stokes equations [130].

2.4.2 Pipe Flow Model

In the cooling system, the water is flowing in circular tubes. The cross section of these tubes and the flow rate are constant. Then, being an incompressible fluid, the fluid mean velocity is also constant.

According to Figure 2.4.1, let r be the radial coordinate, u the axial velocity and u_{max} the velocity at the center line of the tube. We consider a fully developed turbulent flow with constant flow rate, Q , and cross section A_{tube} .

Joining the theoretical work made by Blasius with experimental measurements, Prandtl developed an empirical model which provides good results for high Reynolds turbulent flows (see [164], Chapter 7). This model is commonly called the one-seventh-power law. It assumes that the flow velocity has only the component along the axis of the tube. Then, the flow velocity is axisymmetric with respect to the tube's center line. As a result, it is a function of the mean velocity in the tube's section, u_{avg} , and of the radial distance from the center line only. We summarize the assumptions made:

(H2.1) The axis of the tube is parallel to \mathbf{e}_z .

(H2.2) The flow velocity has only the component along the axis of the tube which is function of the tube radial coordinate only, i.e. $\mathbf{u}(\mathbf{x}) = -u(r)\mathbf{e}_z$.



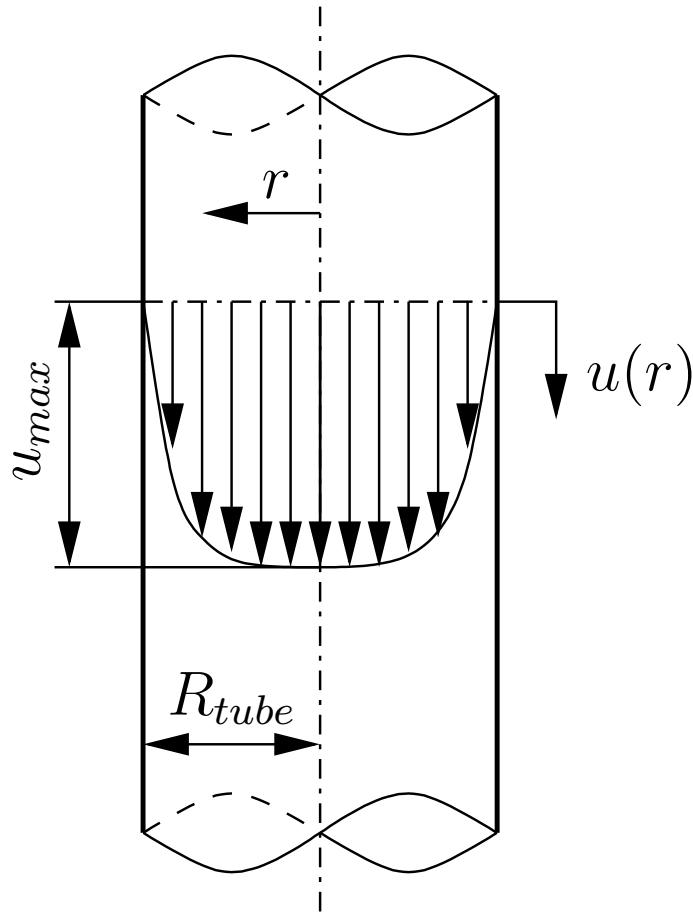


Figure 2.4.1: Schematic of one of the tubes of the cooling system of a CC mold. We denote by R_{tube} the tube radius, r denotes the radial coordinate, u the fluid velocity and u_{max} its maximum value. Notice that in this pipe flow model, the fluid velocity is a function of the radial coordinate only.

The one-seventh-power law developed by Prandtl is

$$u(r) = u_{max} \left(1 - \frac{r}{R}\right)^{\frac{1}{7}}, \quad (2.4.6)$$

The maximum velocity u_{max} can be computed as a function of the Darcy-Weisbach friction factor, f , and the mean tube velocity in the section (see [164])



$$u_{max} = u_{avg}(1 + 1.33\sqrt{f}). \quad (2.4.7)$$

We remember that according to (2.2.1), u_{avg} is constant.

To complete the model, we require an equation to compute the friction factor, f . Several empirical formulas are available. For its simplicity, we use the Swamee-Jain

equation (see [78])

$$f = \frac{0.25}{\left[\log \left(\frac{\epsilon_{Cu}}{3.7} + \frac{5.74}{Re^{0.9}} \right) \right]^2}, \quad (2.4.8)$$

here ϵ_{Cu} is the tube's effective roughness height (1 μm for copper).

2.4.3 Thermal Model

In deriving the thermal model, we make the following assumption:

(H1.9) The fluid velocity, \mathbf{u} , is known, belongs to $V(\Omega_f)$ and is such that $\gamma_{\Gamma_{f_{out}}} \mathbf{u} \cdot \mathbf{n}_f \geq 0$.

This assumption comes from physical consideration on the flow field. In particular, adherence condition to solid walls was considered on Γ_{sf} . Moreover, we assume that at $\Gamma_{f_{out}}$ the flow is directed outward of the domain.

The thermal behavior of the mold, assuming known the hydrodynamic behavior of the fluid, can be modeled by the following advection-diffusion elliptic equations

Problem 2.2. Find T_f and T_s such that

$$\begin{aligned} \rho_f C_{p_f} \mathbf{u}(\mathbf{x}) \cdot \nabla(T_f(\mathbf{x})) - k_f \Delta T_f(\mathbf{x}) &= f_f(\mathbf{x}), & \text{in } \Omega_f, \\ -k_s \Delta T_s(\mathbf{x}) &= f_s(\mathbf{x}), & \text{in } \Omega_s, \end{aligned} \quad (2.4.9)$$

with BCs

$$\begin{cases} T_f(\mathbf{x}) = \tilde{T}_f(\mathbf{x}) & \text{on } \Gamma_{f_{in}}, \end{cases} \quad (2.4.10)$$

$$\begin{cases} -k_f \nabla T_f(\mathbf{x}) \cdot \mathbf{n}_f(\mathbf{x}) = 0 & \text{on } \Gamma_{f_{out}}, \end{cases} \quad (2.4.11)$$

$$\begin{cases} -k_s \nabla T_s(\mathbf{x}) \cdot \mathbf{n}_s(\mathbf{x}) = g(\mathbf{x}) & \text{on } \Gamma_{s_{in}}, \end{cases} \quad (2.4.12)$$

$$\begin{cases} -k_s \nabla T_s(\mathbf{x}) \cdot \mathbf{n}_s(\mathbf{x}) = 0 & \text{on } \Gamma_{s_{ex}}, \end{cases} \quad (2.4.13)$$

and transmission conditions

$$\begin{cases} T_f(\mathbf{x}) = T_s(\mathbf{x}) & \text{on } \Gamma_{sf}, \end{cases} \quad (2.4.14)$$

$$\begin{cases} k_s \nabla T_s(\mathbf{x}) \cdot \mathbf{n}_s(\mathbf{x}) + k_f \nabla T_f(\mathbf{x}) \cdot \mathbf{n}_f(\mathbf{x}) = 0 & \text{on } \Gamma_{sf}. \end{cases} \quad (2.4.15)$$

Below, we try to obtain a weak formulation for this problem. To do this, let us assume for a moment that Problem 2.2 admits a regular solution $T = (T_s, T_f) \in$

$H^1(\Omega)$. So, given a function $z \in H_{\Gamma_{fin}}^1(\Omega)$, we have, by (2.4.9) and integration by parts, that

$$\begin{aligned}
 & \rho_f C_{p_f} \int_{\Omega_f} (\mathbf{u}(\mathbf{x}) \cdot \nabla T_f(\mathbf{x})) z(\mathbf{x}) d\mathbf{x} + \\
 & \int_{\Omega_f} k_f \nabla T_f(\mathbf{x}) \cdot \nabla z(\mathbf{x}) d\mathbf{x} + \int_{\Omega_s} k_s \nabla T_s(\mathbf{x}) \cdot \nabla z(\mathbf{x}) d\mathbf{x} - \\
 & - \int_{\Gamma_f} k_f \frac{\partial T_f(\mathbf{x})}{\partial n} z(\mathbf{x}) d\Gamma - \int_{\Gamma_s} k_s \frac{\partial T_s(\mathbf{x})}{\partial n} z(\mathbf{x}) d\Gamma = \\
 & \int_{\Omega_f} f_f(\mathbf{x}) z(\mathbf{x}) d\mathbf{x} + \int_{\Omega_s} f_s(\mathbf{x}) z(\mathbf{x}) d\mathbf{x}.
 \end{aligned} \tag{2.4.16}$$

Considering boundary conditions (2.4.11)-(2.4.13) and transmission condition (2.4.15), we have

$$\begin{aligned}
 & \rho_f C_{p_f} \int_{\Omega_f} (\mathbf{u}(\mathbf{x}) \cdot \nabla T_f(\mathbf{x})) z_f(\mathbf{x}) d\mathbf{x} + \int_{\Omega_f} k_f \nabla T_f(\mathbf{x}) \cdot \nabla z_f(\mathbf{x}) d\mathbf{x} + \\
 & \int_{\Omega_s} k_s \nabla T_s(\mathbf{x}) \cdot \nabla z_s(\mathbf{x}) d\mathbf{x} + \int_{\Gamma_{sin}} g(\mathbf{x}) z_s(\mathbf{x}) d\Gamma = \\
 & \int_{\Omega_f} f_f(\mathbf{x}) z_f(\mathbf{x}) d\mathbf{x} + \int_{\Omega_s} f_s(\mathbf{x}) z_s(\mathbf{x}) d\mathbf{x}.
 \end{aligned} \tag{2.4.17}$$

From hypothesis (H1.5) and thanks to (2.1.6) and (2.1.7), there exists

$$R_{\tilde{T}} = (R_{\tilde{T}_s}, R_{\tilde{T}_f}) \in H^1(\Omega), \quad \gamma_{\Gamma_{fin}} R_{\tilde{T}} = \tilde{T}_f, \tag{2.4.18}$$

with $\|R_{\tilde{T}}\|_{H^1(\Omega)} \leq C \|\tilde{T}_f\|_{H^{1/2}(\Gamma_{fin})}$. Then, if $T \in H^1(\Omega)$ and verifies (2.4.10), we can write

$$\bar{T} = T - R_{\tilde{T}} \in H_{\Gamma_{fin}}^1(\Omega). \tag{2.4.19}$$

So, we can rewrite (2.4.17) as

$$\begin{aligned}
 & \rho_f C_{p_f} \int_{\Omega_f} (\mathbf{u}(\mathbf{x}) \cdot \nabla \bar{T}_f(\mathbf{x})) z_f(\mathbf{x}) d\mathbf{x} + \int_{\Omega_f} k_f \nabla \bar{T}_f(\mathbf{x}) \cdot \nabla z_f(\mathbf{x}) d\mathbf{x} + \\
 & \int_{\Omega_s} k_s \nabla \bar{T}_s(\mathbf{x}) \cdot \nabla z_s(\mathbf{x}) d\mathbf{x} + \int_{\Gamma_{sin}} g(\mathbf{x}) z_s(\mathbf{x}) d\Gamma = \\
 & \int_{\Omega_f} f_f(\mathbf{x}) z_f(\mathbf{x}) d\mathbf{x} + \int_{\Omega_s} f_s(\mathbf{x}) z_s(\mathbf{x}) d\mathbf{x} - \quad (2.4.20) \\
 & \rho_f C_{p_f} \int_{\Omega_f} (\mathbf{u}(\mathbf{x}) \cdot \nabla R_{\tilde{T}_f}(\mathbf{x})) z_f(\mathbf{x}) d\mathbf{x} - \int_{\Omega_s} k_s \nabla R_{\tilde{T}_s}(\mathbf{x}) \cdot \nabla z_s(\mathbf{x}) d\mathbf{x} - \\
 & \int_{\Omega_f} k_f \nabla R_{\tilde{T}_f}(\mathbf{x}) \cdot \nabla z_f(\mathbf{x}) d\mathbf{x}, \quad \forall z \in H_{\Gamma_{fin}}^1.
 \end{aligned}$$

Given that any regular solution of Problem 2.2 should verify (2.4.20), if it exists, it can be found between the solutions of

Problem 2.3. Find $\bar{T} = (\bar{T}_s, \bar{T}_f) \in H_{\Gamma_{fin}}^1(\Omega)$, such that

$$\begin{aligned}
 & \rho_f C_{p_f} \int_{\Omega_f} (\mathbf{u}(\mathbf{x}) \cdot \nabla \bar{T}_f(\mathbf{x})) z_f(\mathbf{x}) d\mathbf{x} + \int_{\Omega_f} k_f \nabla \bar{T}_f(\mathbf{x}) \cdot \nabla z_f(\mathbf{x}) d\mathbf{x} + \\
 & \int_{\Omega_s} k_s \nabla \bar{T}_s(\mathbf{x}) \cdot \nabla z_s(\mathbf{x}) d\mathbf{x} = - \int_{\Gamma_{sin}} g(\mathbf{x}) z_s(\mathbf{x}) d\Gamma + \\
 & \int_{\Omega_f} f_f(\mathbf{x}) z_f(\mathbf{x}) d\mathbf{x} + \int_{\Omega_s} f_s(\mathbf{x}) z_s(\mathbf{x}) d\mathbf{x} - \quad (2.4.21) \\
 & \int_{\Omega_f} k_f \nabla R_{\tilde{T}_f}(\mathbf{x}) \cdot \nabla z_f(\mathbf{x}) d\mathbf{x} - \rho_f C_{p_f} \int_{\Omega_f} (\mathbf{u}(\mathbf{x}) \cdot \nabla R_{\tilde{T}_f}(\mathbf{x})) z_f(\mathbf{x}) d\mathbf{x} - \\
 & \int_{\Omega_s} k_s \nabla R_{\tilde{T}_s}(\mathbf{x}) \cdot \nabla z_s(\mathbf{x}) d\mathbf{x},
 \end{aligned}$$

for all (z_s, z_f) in $H_{\Gamma_{fin}}^1(\Omega)$.

Notice that all integrals in (2.4.21) are well defined in the spaces considered. Indeed, Problem 2.3 is the proposed weak formulation for Problem 2.2.

Defining the bilinear form

$$\begin{aligned}
 a &: H^1(\Omega) \times H^1(\Omega) \rightarrow \mathbb{R}, \\
 a(w, z) &= \rho_f C_{p_f} \int_{\Omega_f} (\mathbf{u}(\mathbf{x}) \cdot \nabla w_f(\mathbf{x})) z_f(\mathbf{x}) d\mathbf{x} + \\
 &\int_{\Omega_f} k_f \nabla w_f(\mathbf{x}) \cdot \nabla z_f(\mathbf{x}) d\mathbf{x} + \int_{\Omega_s} k_s \nabla w_s(\mathbf{x}) \cdot \nabla z_s(\mathbf{x}) d\mathbf{x},
 \end{aligned} \tag{2.4.22}$$

with w and z belonging to $H^1(\Omega)$; and the linear functional $F : H^1(\Omega) \rightarrow \mathbb{R}$ defined by

$$\begin{aligned}
 F(z) &= - \int_{\Gamma_{s_{in}}} g(\mathbf{x}) z_s(\mathbf{x}) d\Gamma + \int_{\Omega_f} f_f(\mathbf{x}) z_f(\mathbf{x}) d\mathbf{x} + \int_{\Omega_s} f_s(\mathbf{x}) z_s(\mathbf{x}) d\mathbf{x} - \\
 &- \rho_f C_{p_f} \int_{\Omega_f} (\mathbf{u}(\mathbf{x}) \cdot \nabla R_{\bar{T}}(\mathbf{x})) z_f(\mathbf{x}) d\mathbf{x} - \int_{\Omega_f} k_f \nabla R_{\bar{T}}(\mathbf{x}) \cdot \nabla z_f(\mathbf{x}) d\mathbf{x} - \\
 &- \int_{\Omega_s} k_s \nabla R_{\bar{T}_s}(\mathbf{x}) \cdot \nabla z_s(\mathbf{x}) d\mathbf{x},
 \end{aligned} \tag{2.4.23}$$

with z belonging to $H^1(\Omega)$, Problem 2.3 can be written as

$$\begin{aligned}
 \text{Find } \bar{T} &= (\bar{T}_s, \bar{T}_f) \in H_{\Gamma_{fin}}^1(\Omega): a(\bar{T}, z) = F(z), \\
 \forall z &= (z_s, z_f) \in H_{\Gamma_{fin}}^1(\Omega).
 \end{aligned} \tag{2.4.24}$$

We conclude this section by providing the proof of the existence and uniqueness of the solution of Problem 2.3.

Proof. The $H_{\Gamma_{fin}}^1(\Omega)$ -coercivity for the last two terms of (2.4.22) is given thanks to hypothesis (H1.1) (see [5, 50])

$$\begin{aligned}
 &\int_{\Omega_f} k_f \nabla w_f \cdot \nabla w_f d\mathbf{x} + \int_{\Omega_s} k_s \nabla w_s \cdot \nabla w_s d\mathbf{x} = \\
 &k_f \|\nabla w_f\|_{L^2(\Omega_f)}^2 + k_s \|\nabla w_s\|_{L^2(\Omega_s)}^2 \geq \\
 &\min(k_f, k_s) \left(|w_f|_{H^1(\Omega_f)}^2 + |w_s|_{H^1(\Omega_s)}^2 \right) = \\
 &\min(k_f, k_s) |w|_{H^1(\Omega)}^2 \geq \min(k_f, k_s) C_\Omega \|w\|_{H^1(\Omega)}^2,
 \end{aligned} \tag{2.4.25}$$

for all $w = (w_s, w_f) \in H_{\Gamma_{fin}}^1(\Omega)$, where the last inequality is given by the Poincaré inequality, which can be used taking into account that Γ_{fin} has nonzero Hausdorff measure on the boundary; in (2.4.25) C_Ω is a constant only dependent on Ω

(see [124]). We recall that \mathbf{u} belongs to

$$V(\Omega_f) := \{\mathbf{w} \in [H^1(\Omega_f) \cap L^\infty(\Omega_f)]^3 \mid \nabla \cdot \mathbf{w} = 0, \quad \gamma_{\Gamma_{sf}} \mathbf{w} \cdot \mathbf{n}_f = 0\}. \quad (2.4.26)$$

We now verify the $H_{\Gamma_{fin}}^1(\Omega)$ -coercivity of the first term of the bilinear form as defined in (2.4.22)

$$\rho_f C_{pf} \int_{\Omega_f} (\mathbf{u}(\mathbf{x}) \cdot \nabla z_f(\mathbf{x})) z_f(\mathbf{x}) d\mathbf{x} = \frac{1}{2} \rho_f C_{pf} \int_{\Omega_f} [\mathbf{u}(\mathbf{x}) \cdot \nabla (z_f^2(\mathbf{x}))] d\mathbf{x}. \quad (2.4.27)$$

Firstly, notice that all integrals are well defined thanks to $\mathbf{u} \in V(\Omega)$. Integrating by parts, using assumptions (H1.2) and (H1.9), and considering $z_f \in H_{\Gamma_{fin}}^1(\Omega_f)$

$$\begin{aligned} & \frac{1}{2} \rho_f C_{pf} \int_{\Omega_f} (\mathbf{u}(\mathbf{x}) \cdot \nabla z_f^2(\mathbf{x})) d\mathbf{x} = \\ & \frac{1}{2} \rho_f C_{pf} \left(- \int_{\Omega_f} z_f^2(\mathbf{x}) \nabla \cdot \mathbf{u}(\mathbf{x}) d\mathbf{x} + \int_{\Gamma_f} z_f^2(\mathbf{x}) \mathbf{u}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma \right) = \\ & = \frac{1}{2} \rho_f C_{pf} \left(\int_{\Gamma_{fin}} z_f^2(\mathbf{x}) \mathbf{u}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma + \int_{\Gamma_{fout}} z_f^2(\mathbf{x}) \mathbf{u}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma + \right. \\ & \left. + \int_{\Gamma_{fsf}} z_f^2(\mathbf{x}) \mathbf{u}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma \right) = \frac{1}{2} \rho_f C_{pf} \left(\int_{\Gamma_{fout}} z_f^2(\mathbf{x}) \mathbf{u}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma \right) \geq 0. \end{aligned} \quad (2.4.28)$$

The integral $\int_{\Gamma_f} z_f^2(\mathbf{x}) \mathbf{u}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma$ has meaning because the trace of functions of $H^1(\Omega)$ belongs to $L^q(\Gamma)$ for $q = 4$ (see Theorem 4.7 of [102]). The bilinear form is then $H_{\Gamma_{fin}}^1(\Omega)$ -coercive.

For the bilinear form to be continuous must exist a positive constant M such that

$$|a(w, z)| \leq M \|w\|_{H^1(\Omega)} \|z\|_{H^1(\Omega)}, \quad \forall w, z \in H^1(\Omega). \quad (2.4.29)$$

For the first term of (2.4.22), we have

$$\begin{aligned} \left| \rho_f C_{pf} \int_{\Omega_f} (\mathbf{u}(\mathbf{x}) \cdot \nabla w_f(\mathbf{x})) z_f(\mathbf{x}) d\mathbf{x} \right| & \leq \rho_f C_{pf} \|\nabla w_f\|_{L^2(\Omega_f)} \|\mathbf{u}\|_{L^\infty(\Omega_f)} \|z_f\|_{L^2(\Omega_f)} \\ & \leq \rho_f C_{pf} \|w\|_{H^1(\Omega)} \|\mathbf{u}\|_{L^\infty(\Omega_f)} \|z\|_{H^1(\Omega)}, \quad \forall w, z \in H^1(\Omega), \end{aligned} \quad (2.4.30)$$

where Holder and Cauchy-Schwartz inequalities were used as well as the inequality $\|\nabla v\|_{L^2(\Omega)} \leq \|v\|_{H^1(\Omega)}$ for all v in $H^1(\Omega)$. For the second and third term,

$$\begin{aligned} \left| \int_{\Omega_i} k_i \nabla w_i(\mathbf{x}) \cdot \nabla z_i(\mathbf{x}) d\mathbf{x} \right| &\leq k_i \|\nabla w_i\|_{L^2(\Omega_i)} \|\nabla z_i\|_{L^2(\Omega_i)} \\ &\leq k_i \|w\|_{H^1(\Omega)} \|z\|_{H^1(\Omega)}, \quad \forall w, z \in H^1(\Omega) \text{ and } i \in \{s, f\}. \end{aligned} \quad (2.4.31)$$

Thus,

$$|a(w, z)| \leq M \|w\|_{H^1(\Omega)} \|z\|_{H^1(\Omega)}, \quad (2.4.32)$$

with $M = \rho_f C_{p_f} \|\mathbf{u}\|_{L^\infty(\Omega_f)} + k_f + k_s$. Therefore, we have proved that the bilinear form $a(w, z)$ is continuous on $H^1(\Omega) \times H^1(\Omega)$ and $H_{\Gamma_{fin}}^1$ -coercive.

The functional $F(z)$ is clearly linear and bounded by

$$|F(z)| \leq N \|z\|_{H^1(\Omega)}, \quad (2.4.33)$$

with $N = \|g\|_{L^2(\Gamma_{sin})} + \|f_f\|_{L^2(\Omega_f)} + \|f_s\|_{L^2(\Omega_s)} + (\rho_f C_{p_f} \|\mathbf{u}\|_{L^\infty(\Omega_f)} + k_f + k_s) \|R_T\|_{H^1(\Omega)}$.

We can conclude that due to the Lax-Milgram theorem (see [124]), the solution of Problem 2.3 exists and is unique. \square

2.5 THREE-DIMENSIONAL STEADY-STATE HEAT CONDUCTION MODEL COUPLED WITH 1D COOLING MODEL

In this section, we describe the mathematical formulation of model (MS3). This model is inspired by the 2D mold model proposed by Samarasekera and Brimacombe (see [134]). It substitutes the three-dimensional heat transfer problem in the fluid domain with a 1D sectional heat balance equation, imposing convective transmission conditions at the interface and assuming the fluid to be in plug flow.

As already mentioned, the fluid flows in a fully developed turbulent regime. This is due to the high Reynolds number and the steady flow rate (see Section 2.2 and 2.3). Let us consider the value of the Peclet number given in Table 2.1. It is a measure of the quotient of the heat advection with respect to the heat conduction. For this value of the Peclet number, the heat conduction is negligible with respect to the advection in the tube axial direction. Moreover due to the high Reynolds number (2.2.2), the turbulent mixing plays an important role in diffusing temperature radially in each cross section of the channels. Then, more than the fluid temperature field, we are interested in knowing a section averaged value.

To summarize, we add the following assumptions to those of model (MS1)

- (H3.1) The source terms are null: $f_i = 0, i = s, f$.
- (H3.2) The axis of the tube is parallel to \mathbf{e}_z .
- (H3.3) The fluid velocity is known and has only the axial component (plug flow), i.e. $\mathbf{u}(\mathbf{x}) = u\mathbf{e}_z = -u_{avg}\mathbf{e}_z$.
- (H3.4) Due to the high Peclet number, the heat conduction in the tube axial direction is neglected, i.e. $\frac{\partial^2 T}{\partial z^2} = 0$.

Let us denote by Δ_A the Laplacian in a plane perpendicular to the tubes axes. Due to assumptions (H3.1)-(H3.4), the fluid and solid temperature is found by solving

Problem 2.4. Find T_f and T_s such that

$$\begin{aligned} \rho_f C_{p_f} u \frac{\partial T_f(\mathbf{x})}{\partial z} - k_f \Delta_A T_f(\mathbf{x}) &= 0, & \text{in } \Omega_f, \\ -k_s \Delta T_s(\mathbf{x}) &= 0, & \text{in } \Omega_s, \end{aligned} \quad (2.5.1)$$

with BCs

$$\begin{cases} T_f(\mathbf{x}) = \tilde{T}_f(\mathbf{x}) & \text{on } \Gamma_{fin}, \\ -k_f \nabla T_f(\mathbf{x}) \cdot \mathbf{n}_f(\mathbf{x}) = 0 & \text{on } \Gamma_{fout}, \\ -k_s \nabla T_s(\mathbf{x}) \cdot \mathbf{n}_s(\mathbf{x}) = g(\mathbf{x}) & \text{on } \Gamma_{sin}, \\ -k_s \nabla T_s(\mathbf{x}) \cdot \mathbf{n}_s(\mathbf{x}) = 0 & \text{on } \Gamma_{sex} \end{cases} \quad \begin{aligned} (2.5.2) \\ (2.5.3) \\ (2.5.4) \\ (2.5.5) \end{aligned}$$

and transmission conditions

$$\begin{cases} T_f(\mathbf{x}) = T_s(\mathbf{x}) & \text{on } \Gamma_{sf}, \\ k_s \nabla T_s(\mathbf{x}) \cdot \mathbf{n}_s(\mathbf{x}) + k_f \nabla T_f(\mathbf{x}) \cdot \mathbf{n}_f(\mathbf{x}) = 0 & \text{on } \Gamma_{sf}. \end{cases} \quad \begin{aligned} (2.5.6) \\ (2.5.7) \end{aligned}$$

Let us define the section averaged temperature

$$T_{f,avg}(z) := (A_{tube})^{-1} \int_{A_{tube}} T_f(\mathbf{x}) d\sigma, \quad (2.5.8)$$

where the integral is on the normal section of each tube. Integrating the first of the two equations (2.5.1) on the section of each tube, using the Gauss theorem and (2.4.15), we obtain



$$\begin{aligned} \rho_f C_{p_f} \int_{A_{tube}} \frac{\partial T_f}{\partial z} da - \int_{A_{tube}} k_f \Delta_A T_f da &= \\ \rho_f C_{p_f} A_{tube} \frac{dT_{f,avg}}{dz} - \oint_{P_{tube}} k_f \nabla T_f \cdot \mathbf{n}_f dl &= \\ \rho_f C_{p_f} A_{tube} \frac{dT_{f,avg}}{dz} + \oint_{P_{tube}} q_{sf} dl &= 0, \end{aligned} \quad (2.5.9)$$

where P_{tube} is a normal section to Γ_{sf} and $q_{sf} := -k_f \nabla T_f \cdot \mathbf{n}_f = k_s \nabla T_s \cdot \mathbf{n}_s$ is the heat flux at the fluid-solid interface. Then, the mathematical formulation of model (MS3) is

Problem 2.5. Find $T_{f_{avg}}$ and T_s such that

$$\begin{aligned} \rho_f C_{p_f} A_{tube} \frac{dT_{f_{avg}}(z)}{dz} + \oint_{P_{tube}} q_{sf}(\mathbf{x}) dl &= 0, \\ -k_s \Delta T_s(\mathbf{x}) &= 0, \quad \text{in } \Omega_s, \end{aligned} \quad (2.5.10)$$

with BCs

$$\begin{cases} T_{f_{avg}}(z) = \tilde{T}_{f_{avg}} & \text{on } \Gamma_{fin}, \\ -k_s \nabla T_s(\mathbf{x}) \cdot \mathbf{n}_s(\mathbf{x}) = g(\mathbf{x}) & \text{on } \Gamma_{sin}, \\ -k_s \nabla T_s(\mathbf{x}) \cdot \mathbf{n}_s(\mathbf{x}) = 0 & \text{on } \Gamma_{sex} \end{cases} \quad \begin{aligned} (2.5.11) \\ (2.5.12) \\ (2.5.13) \end{aligned}$$

and transmission condition

$$\begin{aligned} T_{f_{avg}}(z) &= \frac{1}{P_{tube}} \oint_{P_{tube}} T_s dl & \text{on } \Gamma_{sf}, \\ k_s \nabla T_s(\mathbf{x}) \cdot \mathbf{n}_s(\mathbf{x}) &= q_{sf}(\mathbf{x}) & \text{on } \Gamma_{sf}, \end{aligned} \quad (2.5.14)$$

where we impose an integral version of the continuity transmission condition on Γ_{sf} .

2.6 THREE-DIMENSIONAL STEADY-STATE HEAT CONDUCTION MODEL

In the present section, we derive from Problem 2.2 the mathematical formulation of model (MS4). It is a threedimensional, steady-state, heat conduction model in the solid domain corresponding to the solid part of the mold.

In this section, we consider the fluid domain to be empty and we model the water cooling by a convection condition on the boundary Γ_{sf} . Then, we apply the following assumptions

(H4.1) The fluid domain is empty: $\Omega_f = \emptyset$.

(H4.2) The heat transfer coefficient is constant and strictly positive: $h \in \mathbb{R}^+$.

(H4.3) The cooling water temperature, T_f , is known, constant, and belongs to $L^2(\Gamma_{sf})$.

These assumptions are motivated by the industrial water temperature measurements. In fact, due to the very high water flow rate, the water temperature increase is of order unity and almost constant along the casting.

Due to assumptions (H4.1)-(H4.3), the thermal model of Problem 2.2 reduces to

Problem 2.6. Find T_s such that

$$-k_s \Delta T_s = 0, \text{ in } \Omega_s, \quad (2.6.1)$$

with BCs

$$\begin{cases} -k_s \nabla T_s \cdot \mathbf{n} = g & \text{on } \Gamma_{s_{in}}, \\ -k_s \nabla T_s \cdot \mathbf{n} = 0 & \text{on } \Gamma_{s_{ex}}, \\ -k_s \nabla T_s \cdot \mathbf{n} = h(T_s - T_f) & \text{on } \Gamma_{sf}. \end{cases} \quad (2.6.2)$$

$$(2.6.3)$$

$$(2.6.4)$$

We recall that for this problem the following result is well established (see [107], Theorem 3.14):

Theorem 2.1. Under the assumptions (H1.4), (H4.2) and (H4.3), the solution to Problem 2.6 exists and is unique in $H^1(\Omega_s)$. Moreover, there exists a $\gamma > 0$ such that the solution to Problem 2.6 belongs to $C^{0,\gamma}(\Omega_s)$.

As a final remark, we recall (see Theorem 3.3.6 in [129]),

Theorem 2.2. If g and T_f belong to $L^s(\Gamma_{s_{in}})$ and $L^s(\Gamma_{sf})$ respectively, with $s > 2$, then the solution T_s to Problem 2.6 belongs to $C(\overline{\Omega_s})$ and

$$\|T_s\|_{C(\overline{\Omega_s})} \leq C \left(\|g\|_{L^s(\Gamma_{s_{in}})} + \|T_f\|_{L^s(\Gamma_{sf})} \right), \quad (2.6.5)$$

where the constant C is independent of h .

2.6.1 Numerical Discretization

This section presents the discretization of the model introduced in the previous section. We use the Finite Volume Method (FVM) for this discretization (see [52]).

In order to perform a space finite volume discretization of Problem 2.6, we introduce a mesh of the domain. In particular, let \mathcal{T} be a mesh on the computational domain Ω_s compatible with the boundary partition. The mesh is such that $\Omega_s = \bigcup_{K \in \mathcal{T}} K$, where an element of \mathcal{T} , denoted by K , is an open subset of Ω_s that we call ‘‘control volume’’.

Due to the geometrical complexity of CC molds, we use unstructured nonorthogonal cell-centered grids. Thus, according to Moraes et al. [96], we can have a second-order accurate finite volume scheme for our diffusion problem.

For the finite volume schemes considered here, the discrete unknown temperature is denoted by T_{s_K} , $K \in \mathcal{T}$. The value T_{s_K} is expected to be some approximation of T_s on the cell K . As usual in FVM, we integrate (2.6.1) over each cell K of the mesh \mathcal{T} . Then, by the divergence formula, we obtain an integral formulation of the fluxes over the boundary of the control volume. Finally, we discretize the fluxes on the boundary with respect to the discrete unknowns.

Let us introduce the following notation. We denote by \mathcal{E} the set of faces of the control volumes. For each $K \in \mathcal{T}$ and $\sigma \in \mathcal{E}$, $m(K)$ denotes the three-dimensional Lebesgue measure of K and $m(\sigma)$ denotes the two-dimensional measure of σ . The mesh size is defined by $\text{size}(\mathcal{T}) := \sup\{m(K), K \in \mathcal{T}\}$. Let N_h be the cardinality of \mathcal{T} . Let $\mathbf{x}_K := \frac{1}{m(K)} \int_K \mathbf{x} d\mathbf{x}$ denote the control volume centroid for each $K \in \mathcal{T}$. Let $\mathbf{x}_\sigma := \frac{1}{m(\sigma)} \int_\sigma \mathbf{x} d\Gamma$ denote the centroid for each $\sigma \in \mathcal{E}$. Moreover, we define $\mathbf{d}_{K,\sigma} := \mathbf{x}_\sigma - \mathbf{x}_K$, $\mathcal{E}_K := \{\sigma \in \mathcal{E} | \sigma \subset \partial K\}$, $\mathcal{E}_{int} := \{\sigma \in \mathcal{E} | \sigma \subset \Omega_s\}$ and $\mathcal{E}_{ext} := \{\sigma \in \mathcal{E} | \sigma \subset \partial\Omega_s\}$.

We start the discretization of Problem 2.6 by integrating (2.6.1) over each control volume and applying the divergence theorem

$$\int_K -k_s \Delta T_s(\mathbf{x}) d\mathbf{x} = \int_{\partial K} -k_s \nabla T_s(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma = 0, \quad \forall K \in \mathcal{T}. \quad (2.6.6)$$

Applying the mean value theorem, we get

$$\int_{\partial K} -k_s \nabla T_s(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma \approx -k_s \sum_{\sigma \in \mathcal{E}_K} (\nabla T_s)_\sigma \cdot \mathbf{s}_\sigma = 0, \quad (2.6.7)$$

where $(\nabla T)_\sigma$ is the gradient evaluated at the face centroid, \mathbf{x}_σ , and $\mathbf{s}_\sigma = \mathbf{n}_\sigma m(\sigma)$, \mathbf{n}_σ being the normal unitary vector, pointing to the outside of K , evaluated at the face centroid. In FVM, we refer to the term $(\nabla T_s)_\sigma \cdot \mathbf{s}_\sigma$ as “flux”.

The difficult in computing the flux at the centroid of the faces of the control volumes is due to the gradient term. Considering the control volume K_P in Figure 2.6.1 and one of its neighbors K_N , we define the vector

$$\mathbf{d}_{PN} := \mathbf{x}_{K_N} - \mathbf{x}_{K_P}. \quad (2.6.8)$$

If the vectors \mathbf{d}_{PN} and \mathbf{s}_σ are aligned for cells sharing the face σ (orthogonal grid), we have the following second order approximation of the gradient evaluated at the cell face

$$(\nabla T_s)_\sigma \cdot \mathbf{s}_\sigma \approx |\mathbf{s}_\sigma| \frac{T_{s_N} - T_{s_P}}{|\mathbf{d}_{PN}|}. \quad (2.6.9)$$

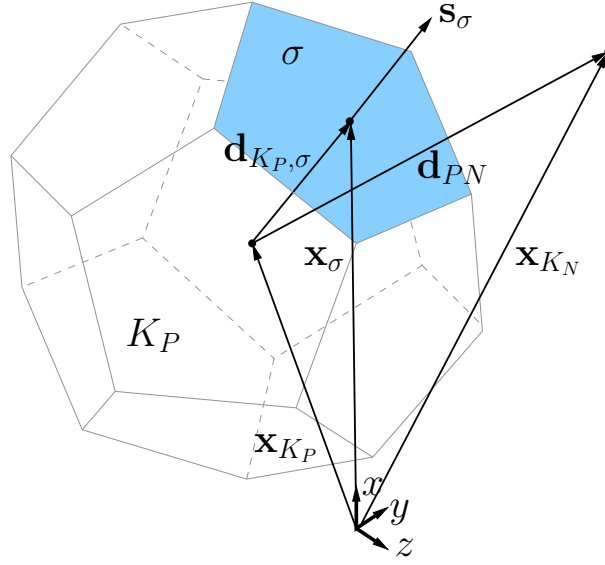


Figure 2.6.1: A polyhedral cell representing the general control volume K_p of the finite volume discretization that is sharing the face σ with its neighbor K_N . We denote by \mathbf{x}_K its centroid, by \mathbf{x}_σ the face centroid, by \mathbf{d}_{PN} the vector going from the centroid of K_P to the centroid of its neighbor, K_N .

We now consider the BCs (2.6.2)-(2.6.4). Let

$$\begin{aligned} g_\sigma &:= \frac{1}{m(\sigma)} \int_\sigma g d\Gamma, \quad \forall \sigma \in \Gamma_{sin}, \\ T_{f_\sigma} &:= \frac{1}{m(\sigma)} \int_\sigma T_f d\Gamma, \quad \forall \sigma \in \Gamma_{sf}. \end{aligned} \quad (2.6.10)$$

Moreover, we introduce similar approximation of T_s at σ , denoted by T_{s_σ} , for all $\sigma \in \mathcal{E}$. Then, we do the following approximations

$$\begin{aligned} (\nabla T_s)_\sigma \cdot \mathbf{s}_\sigma &\approx m(\sigma) \frac{T_{s_\sigma} - T_{s_P}}{|\mathbf{d}_{K_P, \sigma}|}, \quad \forall \sigma \in \Gamma_{sf}, \\ \int_\sigma h(T_s(\mathbf{x}) - T_f(\mathbf{x})) d\Gamma &\approx hm(\sigma)(T_{s_\sigma} - T_{f_\sigma}), \quad \forall \sigma \in \Gamma_{sf}, \\ \int_\sigma g d\Gamma &\approx m(\sigma)g_\sigma, \quad \forall \sigma \in \Gamma_{sin}, \end{aligned} \quad (2.6.11)$$

where K_P is the element at which σ belongs. By introducing the first two equations in (2.6.4), we have

$$-k_s \frac{T_{s_\sigma} - T_{s_P}}{|\mathbf{d}_{K_P, \sigma}|} = h(T_{s_\sigma} - T_{f_\sigma}). \quad (2.6.12)$$

Thus, we can write

$$T_{s_\sigma} = \frac{k_s T_{s_K} - h|\mathbf{d}_{K_P,\sigma}|T_{f_\sigma}}{h|\mathbf{d}_{K_P,\sigma}| + k_s}, \quad \forall \sigma \subset (\Gamma_{sf} \cap \partial K_P). \quad (2.6.13)$$

Then, we have

$$\begin{aligned} -k_s(\nabla T_s)_\sigma \cdot \mathbf{s}_\sigma &\approx -k_s m(\sigma) \frac{T_{s_\sigma} - T_{s_P}}{|\mathbf{d}_{K_P,\sigma}|} \\ &= -\frac{k_s m(\sigma)}{|\mathbf{d}_{K_P,\sigma}|} \left(\frac{k_s}{h|\mathbf{d}_{K_P,\sigma}| + k_s} T_{s_P} + \frac{h|\mathbf{d}_{K_P,\sigma}|}{h|\mathbf{d}_{K_P,\sigma}| + k_s} T_{f_\sigma} - T_{s_P} \right) \\ &= -\frac{k_s m(\sigma)}{|\mathbf{d}_{K_P,\sigma}|} \left(\frac{-h|\mathbf{d}_{K_P,\sigma}|}{h|\mathbf{d}_{K_P,\sigma}| + k_s} T_{s_P} + \frac{h|\mathbf{d}_{K_P,\sigma}|}{h|\mathbf{d}_{K_P,\sigma}| + k_s} T_{f_\sigma} \right). \end{aligned} \quad (2.6.14)$$

Defining

$$q_{K_P,\sigma} := \frac{hk_s m(\sigma)}{h|\mathbf{d}_{K_P,\sigma}| + k_s}, \quad (2.6.15)$$

we can write

$$-k_s(\nabla T_s)_\sigma \cdot \mathbf{s}_\sigma \approx q_{K_P,\sigma}(T_{s_P} - T_{f_\sigma}), \quad \forall \sigma \subset (\Gamma_{sf} \cap \partial K_P). \quad (2.6.16)$$

Then, thanks to (2.6.7)-(2.6.16), we have the following finite volume scheme for Problem 2.6

$$\begin{aligned} &\left(-k_s \sum_{\sigma \in \mathcal{E}_{K_P} \setminus \Gamma} \frac{m(\sigma)}{|\mathbf{d}_{PN}|} + \sum_{\sigma \in \mathcal{E}_{K_P} \cap \Gamma_{sf}} q_{K_P,\sigma} \right) T_{s_P} = \\ &k_s \sum_{\sigma \in \mathcal{E}_{K_P} \setminus \Gamma} \frac{m(\sigma)}{|\mathbf{d}_{PN}|} T_{s_N} + \sum_{\sigma \in \mathcal{E}_{K_P} \cap \Gamma_{sf}} q_{K_P,\sigma} T_{f_\sigma} - \sum_{\sigma \in \mathcal{E}_{K_P} \cap \Gamma_{s_{in}}} m(\sigma) g_\sigma, \quad \forall K_P \in \mathcal{T}. \end{aligned} \quad (2.6.17)$$

As expected by a second-order approximation (see [52]), the first term on the right-hand side depends on the temperature in the neighbors of the cell K . In principles, the discussion is not complete because the general unstructured, non-orthogonal case requires some correction terms to recover the second order accuracy. Then, we refer the interested reader to the work of Moraes et al. [96] for further details.

To conclude, we define the vector of unknowns with dimension N_h

$$(\mathbf{T}_s)_i := T_{s_i}, \quad \forall K_i \in \mathcal{T}, \quad (2.6.18)$$

the matrix $A \in \mathbb{R}^{N_h \times N_h}$, such that

$$\begin{aligned}
 A_{ij} &= -k_s \sum_{\sigma \in \mathcal{E}_{K_i} \setminus \Gamma} \frac{m(\sigma)}{|\mathbf{d}_{PN}|} + \sum_{\sigma \in \mathcal{E}_{K_i} \cap \Gamma_{sf}} q_{K_i, \sigma}, & \text{if } i = j \\
 A_{ij} &= -k_s \sum_{\sigma \in \mathcal{E}_{K_i} \setminus \Gamma} \frac{m(\sigma)}{|\mathbf{d}_{PN}|}, & \text{if } i \text{ neighborhood of } j \\
 A_{ij} &= 0, & \text{otherwise,}
 \end{aligned} \tag{2.6.19}$$

and the source vector

$$(\mathbf{b})_i = \sum_{\sigma \in \mathcal{E}_{K_i} \cap \Gamma_{sf}} q_{K_i, \sigma} T_{f\sigma} - \sum_{\sigma \in \mathcal{E}_{K_i} \cap \Gamma_{sin}} m(\sigma) g_\sigma. \tag{2.6.20}$$

Thus, we can now assemble the linear system related to the scheme (2.6.17)

$$A\mathbf{T}_s = \mathbf{b}. \tag{2.6.21}$$

Since it has the typical structure of a finite volume discretization, we refer to [96] for all the details related to its solution.

3 STEADY-STATE INVERSE PROBLEM

This chapter is devoted to the formulation and solution of the inverse problem of estimating the mold-steel heat flux, g , using the model (MS4) that we derived in Section 2.6.

We formulate two different inverse problems that are differentiated by the available measurement data. Firstly, in Section 3.1, we consider the case of having only the thermocouples temperature measurements. Secondly, in Section 3.2, we assume to have also the total heat flux measurement. In both cases, we initially discuss the mathematical formulation of the inverse problem. Then, we propose some methodologies for its solution.

3.1 INVERSE THREE-DIMENSIONAL STEADY-STATE HEAT CONDUCTION PROBLEM

This section is devoted to the formulation and the study of the inverse problem of Problem 2.6 in the case of having as data the thermocouples temperature measurements. Using a deterministic, least square approach, we state it as an optimal control problem with pointwise observations.

We introduce the following notation. Let $\Psi := \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$ be a collection of points in Ω_s . We define the application $\mathbf{x}_i \in \Psi \rightarrow \hat{T}(\mathbf{x}_i) \in \mathbb{R}^+$, with $\hat{T}(\mathbf{x}_i)$ being the experimentally measured temperature at $\mathbf{x}_i \in \Psi$. Moreover, let G_{ad} be a bounded set in $L^2(\Gamma_{sin})$.

Then, we state the inverse problem as

Problem 3.1. *Given $\{\hat{T}(\mathbf{x}_i)\}_{i=1}^M$, find the heat flux $g \in G_{ad}$ that minimizes the functional $J_1 : L^2(\Gamma_{sin}) \rightarrow \mathbb{R}^+$,*

$$J_1[g] := \frac{1}{2} \sum_{i=1}^M [T_s[g](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)]^2, \quad (3.1.1)$$



where $T_s[g](\mathbf{x}_i)$ is the solution of Problem 2.6 at point \mathbf{x}_i , for all $i = 1, 2, \dots, M$.

Notice that, thanks to Theorem 2.1, the state variable T_s is continuous in Ω_s . Thus, its value at pointwise observations is well-defined.

We now introduce the sensitivity problem related to Problem 2.6. We derive it by perturbing the heat flux $g \rightarrow g + \delta g$, causing a variation of the temperature field, $T_s[g] \rightarrow T_s[g] + \delta T_s[\delta g]$. Subtracting Problem 2.6 from the obtained problem, we have

Problem 3.2. (Sensitivity) Find δT_s such that

$$-k_s \Delta \delta T_s[\delta g] = 0, \quad \text{in } \Omega_s, \quad (3.1.2)$$

with BCs

$$\begin{cases} -k_s \nabla \delta T_s[\delta g] \cdot \mathbf{n} = \delta g & \text{on } \Gamma_{sin}, \\ -k_s \nabla \delta T_s[\delta g] \cdot \mathbf{n} = 0 & \text{on } \Gamma_{sex}, \\ -k_s \nabla \delta T_s[\delta g] \cdot \mathbf{n} = h(\delta T_s[\delta g]) & \text{on } \Gamma_{sf}. \end{cases} \quad (3.1.3)$$

$$\quad \quad \quad (3.1.4)$$

$$\quad \quad \quad (3.1.5)$$

Then,

Proposition 3.1. We have that $T_s[g + \delta g] = T_s[g] + \delta T_s[\delta g]$. Beside, δT_s is linear: $\delta T_s[\delta g_1 + \delta g_2] = \delta T_s[\delta g_1] + \delta T_s[\delta g_2]$.

We now derive in a formal way the adjoint of Problem 3.1. Firstly, we multiply (2.6.1) by a Lagrange multiplier λ . Then, we integrate over Ω_s and add it to (3.1.1) obtaining

$$\mathcal{L}[g, \lambda] = \frac{1}{2} \sum_{i=1}^M (T_s[g](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i))^2 + \int_{\Omega_s} k_s \Delta T_s[g](\mathbf{x}) \lambda(\mathbf{x}) d\mathbf{x}. \quad (3.1.6)$$

To compute the Fréchet derivative with respect to g of $\mathcal{L}[g, \lambda]$, $d\mathcal{L}_g[\delta g, \lambda]$, we first write

$$\begin{aligned} \mathcal{L}[g + \delta g, \lambda] - \mathcal{L}[g, \lambda] &= \sum_{i=1}^M \delta T_s[\delta g](\mathbf{x}_i) (T_s[g](\mathbf{x}_i) + \frac{1}{2} \delta T_s[\delta g](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)) \\ &\quad + \int_{\Omega_s} k_s \lambda(\mathbf{x}) \Delta \delta T_s[\delta g](\mathbf{x}) d\mathbf{x}. \end{aligned} \quad (3.1.7)$$

The Fréchet derivative of \mathcal{L} is then obtained by neglecting the second order terms

$$d\mathcal{L}_g[\delta g, \lambda] = \sum_{i=1}^M \delta T_s[\delta g](\mathbf{x}_i) (T_s[g](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)) + \int_{\Omega_s} k_s \lambda(\mathbf{x}) \Delta \delta T_s[\delta g](\mathbf{x}) d\mathbf{x}. \quad (3.1.8)$$

Finally, Integrating the previous equation twice by parts and applying the BCs of Problem 3.2, we write

$$\begin{aligned}
 d\mathcal{L}_g[\delta g, \lambda] &= \sum_{i=1}^M \delta T_s[\delta g](\mathbf{x}_i)(T_s[g](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)) + \int_{\Omega_s} k_s \Delta \lambda(\mathbf{x}) \delta T_s[\delta g](\mathbf{x}) d\mathbf{x} \\
 &\quad - \int_{\Gamma_{s_{in}} \cup \Gamma_{s_{ex}} \cup \Gamma_{s_{f}}} k_s \delta T_s[\delta g](\mathbf{x}) \nabla \lambda(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma - \int_{\Gamma_{s_{in}}} \lambda(\mathbf{x}) \delta g(\mathbf{x}) d\Gamma \\
 &\quad - \int_{\Gamma_{s_{f}}} h \lambda(\mathbf{x}) \delta T_s[\delta g](\mathbf{x}) d\Gamma.
 \end{aligned} \tag{3.1.9}$$

We can now state the adjoint problem as

Problem 3.3. (*Adjoint*) Find λ such that

$$k_s \Delta \lambda[g] + \sum_{i=1}^M (T_s[g](\mathbf{x}) - \hat{T}(\mathbf{x})) \delta(\mathbf{x} - \mathbf{x}_i) = 0, \quad \text{in } \Omega_s, \tag{3.1.10}$$

with BCs

$$\begin{cases} k_s \nabla \lambda[g] \cdot \mathbf{n} = 0 & \text{on } \Gamma_{s_{in}} \cup \Gamma_{s_{ex}}, \\ k_s \nabla \lambda[g] \cdot \mathbf{n} + h \lambda[g] = 0 & \text{on } \Gamma_{s_{f}}, \end{cases} \tag{3.1.11}$$

$$\tag{3.1.12}$$

$\delta(\mathbf{x} - \mathbf{x}_i)$ being the Dirac function centered at \mathbf{x}_i . A similar problem is proposed in [27] without deriving it.

We notice that if λ is solution of Problem 3.3, $-\lambda[g]$ represents the Fréchet derivative with respect to the inner product in $L^2(\Gamma_{s_{in}})$. Then, we have

$$d\mathcal{L}_g[\delta g, \lambda] = - \int_{\Gamma_{s_{in}}} \lambda[g](\mathbf{x}) \delta g(\mathbf{x}) d\Gamma = \langle -\lambda[g](\mathbf{x}), \delta g(\mathbf{x}) \rangle_{L^2(\Gamma_{s_{in}})}. \tag{3.1.13}$$

Considering that $\mathcal{L}[g, \lambda[g]] = J_1[g]$, the Gâteaux derivative of the functional $J_1[g]$ is



$$J'_{1g} = -\lambda[g] \text{ in } L^2(\Gamma_{s_{in}}). \tag{3.1.14}$$

Different methods can be used for the solution of this minimization problem. Here, we discuss its solution by Alifanov's regularization method (see [2]) and parameterization of the heat flux, g .

3.1.1 Alifanov's Regularization

Alifanov's regularization method is a CGM applied on the adjoint equation (see [99]). We consider the following iterative procedure for the estimation of the function g that minimizes the functional (3.1.1). Given an initial estimation $g^0 \in L^2(\Gamma_{sin})$, for $n > 0$ a new iterant is computed as

$$g^{n+1} = g^n - \beta^n P^n, \quad n = 0, 1, 2, \dots \quad (3.1.15)$$

where n is the iteration counter and β is the stepsize in the search direction P^n given by

$$P^0 = J'_1[g^0], \quad P^{n+1} = J'_1[g^{n+1}] + \gamma^{n+1} P^n \text{ for } n \geq 1, \quad (3.1.16)$$

γ^{n+1} being the conjugate coefficient, and $J'_1[g]$ the Gâteaux derivative of J_1 given by (3.1.14).

The stepsize β^n in (3.1.15) is obtained by minimizing the functional $J_1[g^n - \beta P^n]$ with respect to β . Therefore, β^n is the solution of the critical point equation of the functional J_1 , restricted to a line passing through g^n in the direction defined by P^n , i.e. β^n is the critical point of $J_1[g^n - \beta P^n]$ which then satisfies

$$J_1[g^n - \beta^n P^n] = \min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^M \{T_s[g^n - \beta P^n](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)\}^2 \right\}. \quad (3.1.17)$$

Recalling Problem 3.2,

$$\begin{aligned} J_1[g^n - \beta P^n] &= \frac{1}{2} \sum_{i=1}^M [T_s[g^n - \beta P^n](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)]^2 \\ &= \frac{1}{2} \sum_{i=1}^M [(T_s[g^n] - \beta \delta T_s[P^n])(\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)]^2. \end{aligned} \quad (3.1.18)$$

Differentiating with respect to β , we obtain the critical point equation

$$\frac{dJ_1[g^n - \beta P^n]}{d\beta} = \sum_{i=1}^M [(T_s[g^n] - \beta \delta T_s[P^n])(\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)](-\delta T_s[P^n](\mathbf{x}_i)) = 0. \quad (3.1.19)$$

Finally, we have

$$\beta^n = \frac{\sum_{i=1}^M \{T_s[g^n](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)\} \delta T_s[P^n](\mathbf{x}_i)}{\sum_{i=1}^M (\delta T_s[P^n](\mathbf{x}_i))^2}. \quad (3.1.20)$$

With respect to the conjugate coefficient, γ , its value is zero for the first iteration and for other iterations it can be calculated using Fletcher-Reeves expression as follows (see [56])

$$\gamma^n = \frac{\|J'_1[g^n]\|_{L^2(\Gamma_{sin})}^2}{\|J'_1[g^{n-1}]\|_{L^2(\Gamma_{sin})}^2}. \quad (3.1.21)$$

Notice that, to use this iterative procedure, we have to compute at each iteration the Gâteaux derivative $J'_{1_g}(\mathbf{x})$, which is given by (3.1.14). Thus, we must solve the adjoint problem to compute it.

To conclude, we summarize Alifanov's regularization procedure in Algorithm 1.

Algorithm 1 Alifanov's regularization for the inverse Problem 3.1.

```

Set  $g^0$  and  $n = 0$ 
while  $n < n_{max}$  do
    Compute  $T_s[g^n]$  by solving Problem 2.6
    Compute  $J_1[g^n]$  by (3.1.1)
    if  $J_1[g^n] < J_{1_{tol}}$  then
        Stop
    end if
    Compute  $\lambda[g^n]$  by solving Problem 3.3
    Compute  $J'_1[g^n]$  by (3.1.14)
    if  $n \geq 1$  then
        Compute the conjugate coefficient,  $\gamma^n$ , by (3.1.21)
        Compute the search direction,  $P^n$ , by (3.1.16)
    else
         $P^0 = J'_1[g^0]$ 
    end if
    Compute  $\delta T_s[P^n]$  by solving Problem 3.2 with  $\delta g = P^n$ 
    Compute the stepsize in the search direction,  $\beta^n$ , by (3.1.20)
    Update heat flux  $g^n$  by (3.1.15)
     $n = n + 1$ 
end while
return  $g^n$ 

```

Anticipating the results of Chapter 4, we notice that this algorithm tends to overestimate the heat flux close to the measurement points and underestimates it in between them. It is due to the pointwise source term in the adjoint problem and to the

thermocouples being very close to the boundary. This limitation of this method was already pointed out in the literature by [158]. To overcome it, they proposed the following modification of the method: to compute at each iteration the mean value of the heat flux and use it as initial guess for the next iteration. However, this modification is very artifact and gives only little improvements of the final result. Then, we propose in the next section a novel methodology that exploits the parameterization of the heat flux.

3.1.2 Parameterization of the Boundary Conditions

In this section, we consider a parameterization of the boundary heat flux, g , to find a solution of Problem 3.1 with a direct method. In the literature, the parameterization of g was already proposed (see e.g. [127]). However, we propose a novel approach both for the parameterization and for the solution of the resulting inverse problem.

We start by considering that we want to parameterize an unknown function in $L^2(\Gamma_{s_{in}})$. Then, we notice that in thin slab casting molds, the thermocouples are all located few millimeters inward from $\Gamma_{s_{in}}$ (see Figure 1.1.3). All together they form a uniform 2D grid. Then, to parameterize g , we use Radial Basis Function (RBF) centered at the projections of the thermocouples points on $\Gamma_{s_{in}}$ (see [23]). In particular, we use Gaussian RBF. However, the following discussion can be applied to other basis functions.

The parameterization of the boundary heat flux reads (see [120] appendix)

$$g(\mathbf{x}) \approx \sum_{j=1}^M w_j \phi_j(\mathbf{x}). \quad (3.1.22)$$

where the $\phi_j(\mathbf{x})$ are M known base functions, and the w_j are the respective unknown weights.

Let ξ_i , $1 \leq i \leq M$, be the projection of the point $\mathbf{x}_i \in \Psi$ on $\Gamma_{s_{in}}$, i.e. such that

$$\xi_i = \operatorname{argmin}_{\xi \in \Gamma_{s_{in}}} \|\mathbf{x}_i - \xi\|_2, \quad \mathbf{x}_i \in \Psi. \quad (3.1.23)$$

By centering the RBF in these points, their expression is

$$\phi_j(\mathbf{x}) = e^{-(\eta \|\mathbf{x} - \xi_j\|_2)^2}, \quad \text{for } j = 1, 2, \dots, M, \quad (3.1.24)$$

where η is the shape parameter of the Gaussian basis. By increasing (decreasing) its values, the radial decay of the basis slows down (speeds up).

Notice that by doing this parameterization, we change the problem from estimating a function in an infinite dimensional space to estimating the vector of the coefficients $\mathbf{w} \in \mathbb{R}^M$. Defined the heat flux parameterization, the objective of the inverse problem is to determine \mathbf{w} which identifies g once the basis $\phi_j, j = 1, 2, \dots, M$ are fixed. So, we rewrite the inverse Problem 3.1 as

Problem 3.4. (Inverse) Assume that g is as in (3.1.22). Given the temperature measurements, $\hat{T}(\Psi) \in \mathbb{R}^M$, find $\hat{\mathbf{w}} \in \mathbb{R}^M$ which minimizes the functional

$$J_1[\mathbf{w}] = \frac{1}{2} \sum_{i=1}^M [T_s[g](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)]^2. \quad (3.1.25)$$

To solve Problem 3.4, suppose to have the solutions of Problem 2.6, $T_s[\phi_j]$, for $j = 1, 2, \dots, M$. Denote by T_{ad} the solution of

Problem 3.5. Find T_{ad} such that

$$-k_s \Delta T_{ad} = 0, \quad \text{in } \Omega_s, \quad (3.1.26)$$

with BCs

$$\begin{cases} -k_s \nabla T_{ad} \cdot \mathbf{n} = 0 & \text{on } \Gamma_{sin} \cup \Gamma_{sex}, \\ -k_s \nabla T_{ad} \cdot \mathbf{n} = h(T_{ad} + T_f) & \text{on } \Gamma_{sf}. \end{cases} \quad (3.1.27)$$

$$(3.1.28)$$

We have

Theorem 3.2. Given $\mathbf{w}, T_s[\phi_j]$ for $j = 1, 2, \dots, M$ and T_{ad} ,

$$T_s[\mathbf{w}] = \sum_{j=1}^N w_j (T_s[\phi_j] + T_{ad}) - T_{ad}, \quad (3.1.29)$$

 is the solution to Problem 2.6.

In the following, to simplify notation and if there is no room for error, $T_s[\mathbf{w}]$ represents the solution $T_s[g]$ of Problem 2.6 with g as in (3.1.22).

Proof. By substituting (3.1.29) into (2.6.1), we have

$$-k_s \Delta T_s[\mathbf{w}] = -k_s \Delta \left(\sum_{j=1}^N w_j (T_s[\phi_j] + T_{ad}) - T_{ad} \right) = 0, \quad \text{in } \Omega_s. \quad (3.1.30)$$

On the other hand, with respect to the BCs associated to that problem, we can write

$$\begin{aligned} -k_s \nabla T_s[\mathbf{w}] \cdot \mathbf{n} &= -k_s \nabla \left(\sum_{j=1}^N w_j (T_s[\phi_j] + T_{ad}) - T_{ad} \right) \cdot \mathbf{n} \\ &= \sum_{j=1}^N w_j \phi_j = g, \quad \text{on } \Gamma_{sin} \end{aligned} \quad (3.1.31)$$

$$-k_s \nabla T_s[\mathbf{w}] \cdot \mathbf{n} = -k_s \nabla \left(\sum_{j=1}^N w_j (T_s[\phi_j] + T_{ad}) - T_{ad} \right) \cdot \mathbf{n} = 0, \quad \text{on } \Gamma_{sex} \quad (3.1.32)$$

$$\begin{aligned} -k_s \nabla T_s[\mathbf{w}] \cdot \mathbf{n} &= -k_s \nabla \left(\sum_{j=1}^N w_j (T_s[\phi_j] + T_{ad}) - T_{ad} \right) \cdot \mathbf{n} \\ &= h \left(\sum_{j=1}^N w_j (T_s[\phi_j] + T_{ad}) - T_{ad} - T_f \right), \quad \text{on } \Gamma_{sf} \end{aligned} \quad (3.1.33)$$

□

Let us now define the residual $\mathbf{R}[\mathbf{w}] \in \mathbb{R}^M$ as the vector whose components are

$$(\mathbf{R}[\mathbf{w}])_i := (\mathbf{T}[\mathbf{w}])_i - (\hat{\mathbf{T}})_i, \quad (3.1.34)$$

where $\mathbf{T}[\mathbf{w}]$ and $\hat{\mathbf{T}}$ denote the vectors of \mathbb{R}^M whose i -components are $(\mathbf{T}[\mathbf{w}])_i = T_s[\mathbf{w}](\mathbf{x}_i)$ and $(\hat{\mathbf{T}})_i = \hat{T}(\mathbf{x}_i)$, respectively. So, we can rewrite the cost functional of the inverse problem (3.1.25) as

$$J_1[\mathbf{w}] = \frac{1}{2} \mathbf{R}[\mathbf{w}]^T \mathbf{R}[\mathbf{w}]. \quad (3.1.35)$$

To minimize it, we solve the critical point equation

$$\frac{\partial J_1[\hat{\mathbf{w}}]}{\partial w_j} = \sum_{i=1}^M R[\hat{\mathbf{w}}]_i \frac{\partial (T_s[\hat{\mathbf{w}}])_i}{\partial w_j} = 0, \quad \text{for } j = 1, 2, \dots, M. \quad (3.1.36)$$

Thanks to Theorem 3.2, equation (3.1.36) can be written as

$$\mathbf{R}[\hat{\mathbf{w}}]^T (\mathbf{T}[\phi_j] + \mathbf{T}_{ad}) = 0, \text{ for } j = 1, 2, \dots, M, \quad (3.1.37)$$

being \mathbf{T}_{ad} the vector of \mathbb{R}^M whose i -component is $T_{ad}(\mathbf{x}_i)$. Then, the vector associated to the solution of the direct problem in the measurement points, $\mathbf{T}[\mathbf{w}] \in \mathbb{R}^M$, is given by

$$\mathbf{T}[\mathbf{w}] = \sum_{j=1}^M w_j \mathbf{T}[\phi_j] + \left(\sum_{j=1}^M w_j - 1 \right) \mathbf{T}_{ad}. \quad (3.1.38)$$

Let us denote by Θ the matrix of $\mathbb{R}^{M \times M}$ such that

$$\Theta_{ij} := T_s[\phi_j](\mathbf{x}_i) + T_{ad}(\mathbf{x}_i). \quad (3.1.39)$$

Then, (3.1.37) can be written as

$$\Theta^T \mathbf{R}[\hat{\mathbf{w}}] = 0. \quad (3.1.40)$$

Recalling the definition of \mathbf{R} and (3.1.38), we have

$$\Theta^T \mathbf{R}[\hat{\mathbf{w}}] = \Theta^T (\Theta \hat{\mathbf{w}} - \mathbf{T}_{ad} - \hat{\mathbf{T}}) = 0. \quad (3.1.41)$$

Then, we obtain the solution of the inverse problem by solving the linear system of order $M \times M$

$$\Theta^T \Theta \hat{\mathbf{w}} = \Theta^T (\hat{\mathbf{T}} + \mathbf{T}_{ad}). \quad (3.1.42)$$

In the literature, this is generally called the normal equation.

In this setting, (3.1.42) is a linear map from the observations to the heat flux weights. Consequently, we have that the existence and uniqueness of the solution to the inverse problem depends on the invertibility of the matrix $\Theta^T \Theta$. We can easily see that the matrix $\Theta^T \Theta$ is symmetric and positive semi-definite. In general however, we can not ensure that it is invertible. In fact, the invertibility depends on the choice of the basis function, the computational domain and the BCs.

In the numerical tests, we will see that this matrix tends to be ill-conditioned. This is a reflect of the ill-posedness of the inverse problem. Different regularization techniques for linear systems are available to overcome this issue (see e.g. [7]). Here, we consider the Truncated Singular Value Decomposition (TSVD) regularization.

To briefly describe this regularization technique, we denote the Singular Values Decomposition (SVD) of $\Theta^T \Theta$ by

$$\Theta^T \Theta = U \Sigma V^T = \sum_{i=1}^r \mathbf{u}_i \sigma_i \mathbf{v}_i^T, \quad (3.1.43)$$

where σ_i denotes the i -th singular value of $\Theta^T \Theta$ (numbered according to their decreasing value), r denotes the last non null singular value (i.e. the rank of $\Theta^T \Theta$), \mathbf{u}_i and \mathbf{v}_i are the i -th column of the semi-unitary matrices U and V respectively (both belonging to $\mathbb{M}^{M \times r}$), and Σ is the square matrix of $\mathbb{M}^{r \times r}$ such that $\Sigma_{ii} = \sigma_i$ and $\Sigma_{ij} = 0$ if $i \neq j$. Then, the TSVD regularized solution of (3.1.42) is

$$\mathbf{w} = \sum_{i=1}^{\alpha_{TSVD}} \left(\frac{\mathbf{u}_i^T \Theta^T (\hat{\mathbf{T}} + \mathbf{T}_{ad})}{\sigma_i} \right) \mathbf{v}_i. \quad (3.1.44)$$

This solution differs from the least square solution only in that the sum is truncated at $i = \alpha_{TSVD}$ instead of $i = r$. In this way, we cut off the smallest singular values that are responsible of the errors propagation. For a detailed discussion on the solution of discrete ill-posed inverse problems, we refer the reader to Hansen's monograph on the subject [65].

We conclude our discussion of this method by noticing its most interesting feature for our investigation. In fact, it is already suitable for real-time computations since we can divide it into an offline (expensive) phase and an online (cheap) phase. In the offline phase, we compute the $T_s[\phi_j]$ for $j = 1, 2, \dots, M$ and T_{ad} by solving Problem 2.6 and Problem 3.5. Then, in the online phase, we input the measurements $\hat{\mathbf{T}}$ and solve the linear system (3.1.42). For the choice made when selecting the basis functions, the linear system has the dimensions of the number of thermocouples. As a consequence, we can solve it in real-time even with limited computational power. This makes this method very promising for our application.

As a final remark, we notice that for the application of this method, linearity of the direct problem is essential. In fact, it is a necessary condition for Theorem 3.2.

3.2 INVERSE THREE-DIMENSIONAL STEADY-STATE HEAT CONDUCTION PROBLEM WITH TOTAL HEAT FLUX MEASURE

In CC molds, together with the thermocouples pointwise measurements, we can also have approximated total heat flux measurements. In fact, assuming all boundaries but Γ_{sin} and Γ_{sf} to be adiabatic, all heat is extracted by the cooling water at Γ_{sf} . Further, assuming the water heat capacity, C_{pf} , constant, the water mass flow rate, \dot{m} , known and the process to be at steady-state, the total heat flux is given by

$$\hat{G} = \int_{\Gamma_{sin}} g d\Gamma = \dot{m} C_{pf} (T_{f_{out}} - T_{f_{in}}), \quad (3.2.1)$$

where $T_{f_{in}}$ and $T_{f_{out}}$ are the cooling water temperatures at the inlet and outlet of the cooling system, respectively. Then, the total heat flux measurements are obtained by equation (3.2.1) where $T_{f_{in}}$, $T_{f_{out}}$ and the water mass flow rate \dot{m} are experimentally measured.

In this section, we discuss the formulation and solution of the inverse problem of estimating the boundary heat flux, g , by considering both the thermocouples and total heat flux measurements.

Using again a least square, deterministic approach, we state the inverse problem as

Problem 3.6. (Inverse) Given $\{\hat{T}(\mathbf{x}_i)\}_{i=1}^M$ and \hat{G} , find the heat flux $g \in G_{ad}$ that minimizes the functional $J_2 : L^2(\Gamma_{s_{in}}) \rightarrow \mathbb{R}^+$,

$$J_2[g] := \frac{1}{2} \sum_{i=1}^M [T_s[g](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)]^2 + \frac{1}{2} p_g \left(\int_{\Gamma_{s_{in}}} g d\Gamma - \hat{G} \right)^2, \quad (3.2.2)$$

where $T_s[g](\mathbf{x}_i)$ is the solution of Problem 2.6 at points \mathbf{x}_i , for all $i = 1, 2, \dots, M$, and $p_g [\frac{K^2}{W^2}]$ is a weight applied to the total heat measurement.

Notice that, thanks to Theorem 2.1 the state variable T_s is continuous in Ω_s , then its value at pointwise observations is well-defined.

To derive the adjoint of Problem 3.6, we redo computations (3.1.6)-(3.1.9). It turns out that it is again Problem 3.3. However, the Fréchet derivative with respect to the inner product in $L^2(\Gamma_{s_{in}})$ of J_2 is

$$\begin{aligned} d\mathcal{L}_g[\delta g, \lambda] &= - \int_{\Gamma_{s_{in}}} \left[\lambda[g](\mathbf{x}) - p_g \left(\int_{\Gamma_{s_{in}}} g d\Gamma - \hat{G} \right) \right] \delta g(\mathbf{x}) d\Gamma = \\ & \langle -\lambda[g] + p_g \left(\int_{\Gamma_{s_{in}}} g d\Gamma - \hat{G} \right), \delta g \rangle_{L^2(\Gamma_{s_{in}})}. \end{aligned} \quad (3.2.3)$$

Considering that $\mathcal{L}[g, \lambda[g]] = J_2[g]$, the Gâteaux derivative of the functional $J_2[g]$ is

$$J_2'[g] = -\lambda[g] + p_g \left(\int_{\Gamma_{s_{in}}} g d\Gamma - \hat{G} \right) \text{ in } L^2(\Gamma_{s_{in}}). \quad (3.2.4)$$

Different methods can be used for the solution of this minimization problem. As for the minimization of J_1 , we discuss its solution by Alifanov's regularization method and by parameterization of the heat flux, g .

3.2.1 Alifanov's Regularization

In this section, we expand the discussion in Section 3.1.1 on Alifanov's regularization to the inverse Problem 3.6.

We consider the following iterative procedure for the estimation of the function g that minimizes functional (3.1.1). Given an initial estimation $g^0 \in L^2(\Gamma_{s_{in}})$, for $n > 0$ a new iterant is computed by (3.1.15), with the conjugate direction given by

$$P^0 = J_2'[g^0], \quad P^{n+1} = J_2'[g^{n+1}] + \gamma^{n+1}P^n \text{ for } n \geq 1, \quad (3.2.5)$$

γ^n being the conjugate coefficient, and $J_2'[g]$ the Gâteaux derivative of J_2 .

The stepsize, β^n , is obtained by minimizing the functional $J_2[g^n - \beta P^n]$ with respect to β . Therefore, β^n is the solution of the critical point equation of the functional J , restricted to a line passing through g^n in the direction defined by P^n , i.e. β^n is the critical point of $J_2[g^n - \beta P^n]$ which then satisfies

$$J_2[g^n - \beta^n P^n] = \min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^M \{T_s[g^n - \beta P^n](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)\}^2 + \frac{1}{2} p_g \left(\int_{\Gamma_{s_{in}}} g^n - \beta^n P^n d\Gamma - \hat{G} \right)^2 \right\}. \quad (3.2.6)$$

Recalling Problem 3.2,

$$\begin{aligned} J_2[g^n - \beta P^n] &= \\ \frac{1}{2} \sum_{i=1}^M [T_s[g^n - \beta P^n](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)]^2 + \frac{1}{2} p_g \left(\int_{\Gamma_{s_{in}}} (g^n - \beta^n P^n) d\Gamma - \hat{G} \right)^2 &= \\ \frac{1}{2} \sum_{i=1}^M [(T_s[g^n] - \beta \delta T[P^n])(\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)]^2 + \frac{1}{2} p_g \left(\int_{\Gamma_{s_{in}}} (g^n - \beta^n P^n) d\Gamma - \hat{G} \right)^2. & \end{aligned} \quad (3.2.7)$$

Differentiating with respect to β , we obtain the critical point equation

$$\begin{aligned} \frac{dJ_2[g^n - \beta^n P^n]}{d\beta} &= \sum_{i=1}^M [(T_s[g^n] - \beta^n \delta T[P^n])(\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)](-\delta T[P^n](\mathbf{x}_i)) + \\ p_g \left(\int_{\Gamma_{s_{in}}} (g^n - \beta^n P^n) d\Gamma - \hat{G} \right) \left(- \int_{\Gamma_{s_{in}}} P^n d\Gamma \right) &= 0. \end{aligned} \quad (3.2.8)$$

Finally, we have

$$\beta^n = \frac{\sum_{i=1}^M \{T_s[g^n](\mathbf{x}_i) - \hat{T}(\mathbf{x}_i)\} \delta T[P^n](\mathbf{x}_i)}{\sum_{i=1}^M (\delta T_s[P^n](\mathbf{x}_i))^2 - p_g \left(\int_{\Gamma_{s_{in}}} P^n d\Gamma \right)^2} - \frac{p_g \left(\int_{\Gamma_{s_{in}}} P^n d\Gamma \right) \left(\int_{\Gamma_{s_{in}}} g^n d\Gamma - \hat{G} \right)}{\sum_{i=1}^M (\delta T_s[P^n](\mathbf{x}_i))^2 - p_g \left(\int_{\Gamma_{s_{in}}} P^n d\Gamma \right)^2}. \quad (3.2.9)$$

With respect to the conjugate coefficient, γ , its value is zero for the first iteration and for other iterations it can be calculated using Fletcher-Reeves expression as follows [56]

$$\gamma^n = \frac{\|J'_2[g^n]\|_{L^2(\Gamma_{s_{in}})}^2}{\|J'_2[g^{n-1}]\|_{L^2(\Gamma_{s_{in}})}^2}. \quad (3.2.10)$$

Notice that, to use this iterative procedure, we have to compute at each iteration the Gâteaux derivative $J'_{2_g}(\mathbf{x})$ which is given by (3.2.4). Thus, we must solve the adjoint problem to compute it.

Alifanov's regularization algorithm for this inverse problem is summarized in Algorithm 2.

3.2.2 Parameterization of the Boundary Conditions

In this section, we apply the discussion made in Section 3.1.2 to Problem 3.6. Considering the parameterization (3.1.22), due to (3.1.34), we rewrite (3.2.2) as

$$\begin{aligned} J_2[\mathbf{w}] &= \frac{1}{2} \mathbf{R}^T \mathbf{R} + \frac{1}{2} p_g \left(\int_{\Gamma_{s_{in}}} \sum_{j=1}^N w_j \phi_j(\mathbf{x}) d\Gamma - \hat{G} \right)^2 \\ &= \frac{1}{2} \mathbf{R}^T \mathbf{R} + \frac{1}{2} p_g \left(\sum_{j=1}^N w_j \int_{\Gamma_{s_{in}}} \phi_j(\mathbf{x}) d\Gamma - \hat{G} \right)^2. \end{aligned} \quad (3.2.11)$$



Defining the vector in \mathbb{R}^N such that

$$(\boldsymbol{\phi})_i := \int_{\Gamma_{s_{in}}} \phi_j(\mathbf{x}) d\Gamma, \quad (3.2.12)$$

Algorithm 2 Alifanov's regularization with total heat flux measurement for solving inverse Problem 3.6.

Set g^0 and $n = 0$
while $n < n_{max}$ **do**
 Compute $T_s[g^n]$ by solving Problem 2.6
 Compute $J_2[g^n]$ by (3.2.2)
 if $J_2[g^n] < J_{2tol}$ **then**
 Stop
 end if
 Compute $\lambda[g^n]$ by solving Problem 3.3
 Compute $J'_2[g^n]$ by (3.2.4)
 if $n \geq 1$ **then**
 Compute the conjugate coefficient, γ^n , by (3.2.10)
 Compute the search direction, $P^n(\mathbf{x})$, by (3.2.5)
 else
 $P^0 = J'_2[g^0]$
 end if
 Compute $\delta T_s[P^n]$ by solving Problem 3.2 with $\delta g = P^n$
 Compute the stepsize in the search direction, β^n , by (3.2.9)
 Update heat flux g^n by (3.1.15)
 $n = n + 1$
end while
return g^n

we write

$$J_2[\mathbf{w}] = \frac{1}{2} \mathbf{R}^T \mathbf{R} + \frac{1}{2} p_g (\mathbf{w}^T \boldsymbol{\phi} - G)^2. \quad (3.2.13)$$

As in Section 3.1.2, we now write the critical point equation for $J_2[\mathbf{w}]$

$$\frac{\partial J_2[\hat{\mathbf{w}}]}{\partial w_j} = \sum_{i=1}^M R_i \frac{\partial (T_s[\hat{\mathbf{w}}])_i}{\partial w_j} + p_g (\hat{\mathbf{w}}^T \boldsymbol{\phi} - G) (\boldsymbol{\phi})_j = 0, \quad \text{for } j = 1, 2, \dots, N. \quad (3.2.14)$$

Then, introducing the matrix in $\mathbb{R}^{N \times N}$ such that

$$\Phi_{ij} = (\boldsymbol{\phi})_i (\boldsymbol{\phi})_j, \quad (3.2.15)$$

we can write the critical point equation as

$$(\Theta^T \Theta + p_g \Phi) \hat{\mathbf{w}} = p_g G \boldsymbol{\phi} + \Theta^T (\mathbf{T}_{ad} + \hat{\mathbf{T}}). \quad (3.2.16)$$

By solving the linear system (3.2.16) we obtain the weights $\hat{\mathbf{w}}$ of the parameterization. Then, by (3.1.22) we compute the estimated heat flux g . Also in this setting, the discussion at the end of Section 3.1.2 on the regularization of the linear system and the offline-online decomposition holds.

4 BENCHMARKS

In this chapter, we present several numerical tests conducted on designed benchmark cases. It is through these tests that we validate and analyze the performances of the direct and inverse problems solution methodologies that we proposed in the previous sections.

4.1 BENCHMARK 1

In this section, we propose an academic benchmark case. It is a steady-state heat conduction problem in a homogeneous isotropic solid occupying a rectangular parallelepiped domain. By carefully selecting the BCs on the faces of the parallelepiped, we are able to compute the analytical solution of the heat conduction problem. Then, we use this academic test to validate the numerical solution of the direct problem. Moreover, by arbitrarily selecting some temperature measurements points, we test the different inverse problem solution methodologies discussed in Chapter 3.

Let the domain be $\Omega_s = (0, L) \times (0, W) \times (0, H)$ as in Figure 4.1.1 with positive real constants L, W and H . Let Γ_s be boundary of Ω_s . Then, the different boundaries of the domain to be considered are

$$\begin{aligned} \Gamma_{sf} &:= \{\mathbf{x} \in \Gamma_s \mid \mathbf{x} = (x, W, z)\}, & \Gamma_{sin} &:= \{\mathbf{x} \in \Gamma_s \mid \mathbf{x} = (x, 0, z)\}, \\ \Gamma_I &:= \{\mathbf{x} \in \Gamma_s \mid \mathbf{x} = (x, y, H)\}, & \Gamma_{III} &:= \{\mathbf{x} \in \Gamma_s \mid \mathbf{x} = (x, y, 0)\}, \\ \Gamma_{II} &:= \{\mathbf{x} \in \Gamma_s \mid \mathbf{x} = (L, y, z)\}, & \Gamma_{IV} &:= \{\mathbf{x} \in \Gamma_s \mid \mathbf{x} = (0, y, z)\}. \end{aligned} \quad (4.1.1)$$

To have an analytical solution, T_{an} , in Ω_s , we consider a slight modification of Problem 2.6 that does not change its essential aspects.

Problem 4.1. Find T_s such that

$$-k_s \Delta T_s = 0, \text{ in } \Omega_s, \quad (4.1.2)$$

with BCs

$$\left\{ \begin{array}{ll} -k_s \nabla T_s \cdot \mathbf{n} = g_{an} & \text{on } \Gamma_{sin}, \\ -k_s \nabla T_s \cdot \mathbf{n} = q_L & \text{on } \Gamma_L, L \in \{I, II, III, IV\}, \\ -k_s \nabla T_s \cdot \mathbf{n} = h(T_s - T_f) & \text{on } \Gamma_{sf}. \end{array} \right. \quad (4.1.3)$$

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

$$(4.1.4)$$

$$(4.1.5)$$

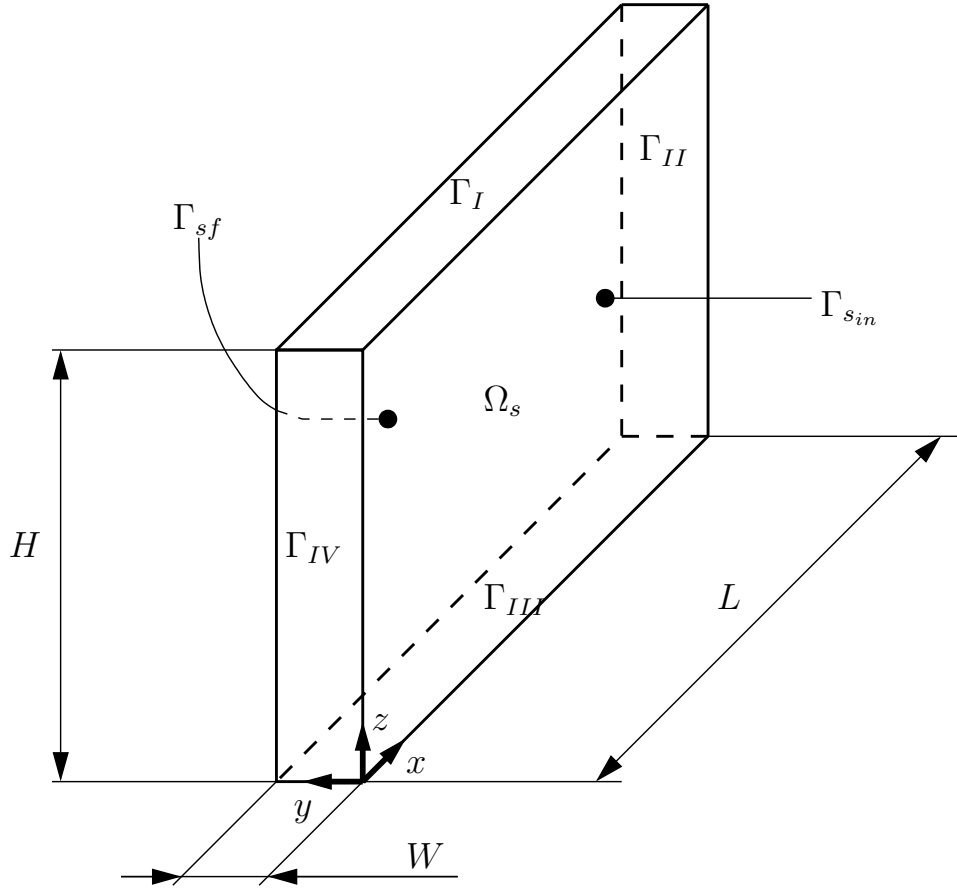


Figure 4.1.1: Schematic of the solid rectangular parallelepiped domain that we use in the benchmark test cases.

Let a, b, c be real constants. To have an analytical solution in Ω_s , we consider the following data as BCs for Problem 4.1,

$$\begin{aligned}
 q_I(\mathbf{x}) &= 2k_s a H, & q_{III}(\mathbf{x}) &= 0, \\
 q_{II}(\mathbf{x}) &= -k_s(2aL + by), & q_{IV}(\mathbf{x}) &= k_s by, \\
 T_f(\mathbf{x}) &= \frac{k_s(bx + c)}{h} + ax^2 + cy - az^2 + bxW + c,
 \end{aligned} \tag{4.1.6}$$

with

$$g_{an}(\mathbf{x}) = k_s(bx + c), \tag{4.1.7}$$

k_s being the thermal considered conductivity, that is assumed constant. Then,

$$T_{an}(\mathbf{x}) = ax^2 + bxy + cy - az^2 + c, \tag{4.1.8}$$

is the solution to Problem 4.1.

4.1.1 Direct Problem

We now discuss the numerical solution of Problem 4.1. Due to its simplicity, the domain Ω_s is discretized by uniform, structured, orthogonal, hexahedral meshes. To study the convergence of the numerical solution to the analytical one, we consider grids with different degree of refinement. In all tests, we use the same number of edges for the three axes.

As previously discussed in Section 2.6.1, we use the FVM for the discretization of Problem 4.1. With respect to the used finite volume scheme, since we have a structured orthogonal grid, no correction is needed when computing the gradient normal to the cells faces. Moreover, we use linear interpolation to interpolate the values from cell centers to face centers. The resulting scheme is second order accurate.

From the discretization of Problem 4.1, we obtain a linear system as in (2.6.21). We solve it by using a preconditioned CG solver with diagonal incomplete Cholesky preconditioning. The tolerance used for the linear system solver is 10^{-12} . All the computations are performed in ITHACA-FV [73, 142] which is a C++ library based on OpenFOAM [98] developed at the SISSA Mathlab.

Finally, Table 4.1 summarizes the parameters used for the computations.

Table 4.1: Parameters used for the simulation of Benchmark 1.

Parameter	Value
Thermal conductivity, k	$3.0 \text{ W}/(mK)$
Heat transfer coefficient, h	$5.0 \text{ W}/(m^2K)$
a	$5 \text{ K}/m^2$
b	$10 \text{ K}/m^2$
c	$15 \text{ K}/m^2$
L	1 m
W	1 m
H	1 m



To evaluate the accuracy of the numerical solutions, we show in Figure 4.1.2 the decay of the absolute and relative difference in the L^2 -norm between the computed and true temperature field. The test confirms the second order accuracy of this finite volume scheme. We conclude that Problem 4.1 is numerically well solved.

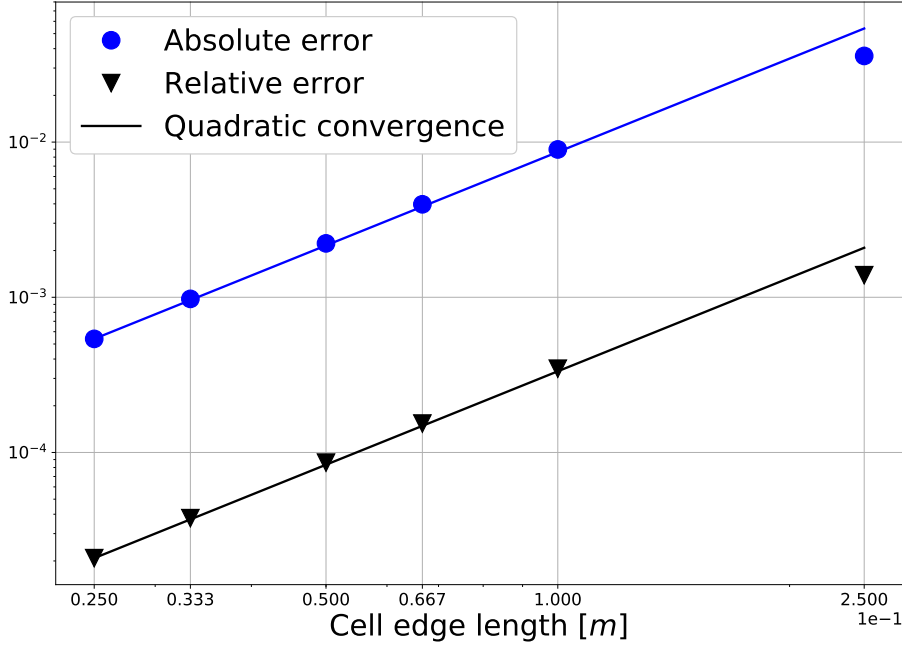


Figure 4.1.2: Decay of the absolute and relative difference between the L^2 -norm of the computed and true temperature field with the mesh refinement.

4.1.2 Inverse Problem with Temperature Measurements

To numerically analyze the performances of the inverse solvers, we design the following test: we select a surface inside Ω_s which is parallel to $\Gamma_{s_{in}}$, and on this surface we locate M measurement points which correspond to the location of M virtual thermocouples. The temperature in these points is given by $\hat{T}(\mathbf{x}_i) = T_{an}(\mathbf{x}_i)$, $i = 1, \dots, M$, being T_{an} the solution of Problem 4.1, given by (4.1.8). Using these temperatures as measurements, we apply the methods described in Chapter 3 to solve the inverse Problem 3.1, considering $T_s[g](\mathbf{x}_i)$ as the solution of Problem 4.1 replacing g_{an} by g .

The virtual thermocouples are located in the plane $y = 0.2$ m. Their (x, z) coordinates are shown in Figure 4.1.3. Then, we have 16 thermocouples located on the nodes of a uniform lattice at the plane $y = 0.2$ m, unless otherwise stated.

The parameters used for the computations are summarized in Table 4.2. In this section, we test the inverse methodologies of Chapter 3 analyzing the effect of different parameters such as grid refinement, CG stopping criterion, RBF shape parameter, measurement noise, etc. To analyze the numerical results, we will often use the fol-

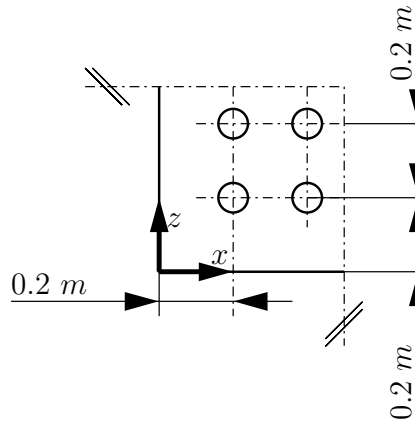


Figure 4.1.3: Positions of the virtual thermocouples for Benchmark 1.

lowing error norms

$$\|\varepsilon\|_{L^2(\Gamma_{sin})} = \left\| \frac{g - g_{an}}{g_{an}} \right\|_{L^2(\Gamma_{sin})}, \quad \|\varepsilon\|_{L^\infty(\Gamma_{sin})} = \left\| \frac{g - g_{an}}{g_{an}} \right\|_{L^\infty(\Gamma_{sin})}. \quad (4.1.9)$$

Notice that from (4.1.7), $g_{an} > 0$.

Table 4.2: Parameters used in testing the inverse problem solvers in Benchmark 1.

Parameter	Value
N. of thermocouples	16
Thermocouples plane	$y = 0.2 \text{ m}$
g^0	0 W/m^2
RBF kernel	Gaussian
N. of RBF	16
Shape parameter, η	0.7

4.1.2.1 Alifanov's Regularization

Now, we analyze the effect that the grid refinement and the stop criterion have on the results obtained by Alifanov's regularization.

We begin by comparing in Figure 4.1.4 (a) and (b) the behavior of the functional J_1 together with the L^2 - and L^∞ -norm of the relative error defined in (4.1.9) as functions of the number of iterations of Algorithm 1. Both the cost function and the

relative error have a sharp decay in the first 10 iterations. Then, the convergence rate has a dramatic decrease reaching a plateau after 60 iterations.

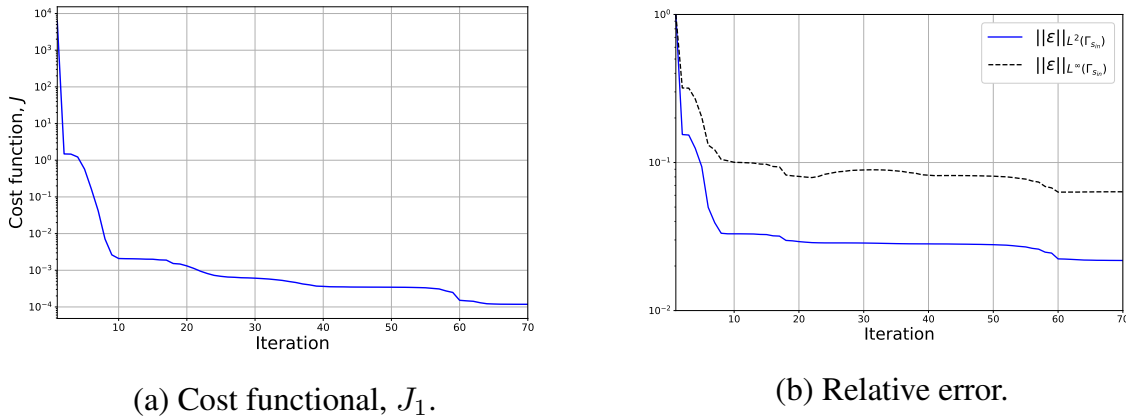


Figure 4.1.4: Behavior of the cost functional J_1 (a) and of the heat flux relative error L^2 - and L^∞ -norms (b) as a function of Alifanov’s regularization iterations for the Benchmark 1.

To have qualitatively insight on the results, we compare the computed heat flux at different iterations in Figure 4.1.5. In few iterations, the estimated heat flux is already in good agreement with the analytical BC and iterating further only provides a slightly improvement of the estimation.

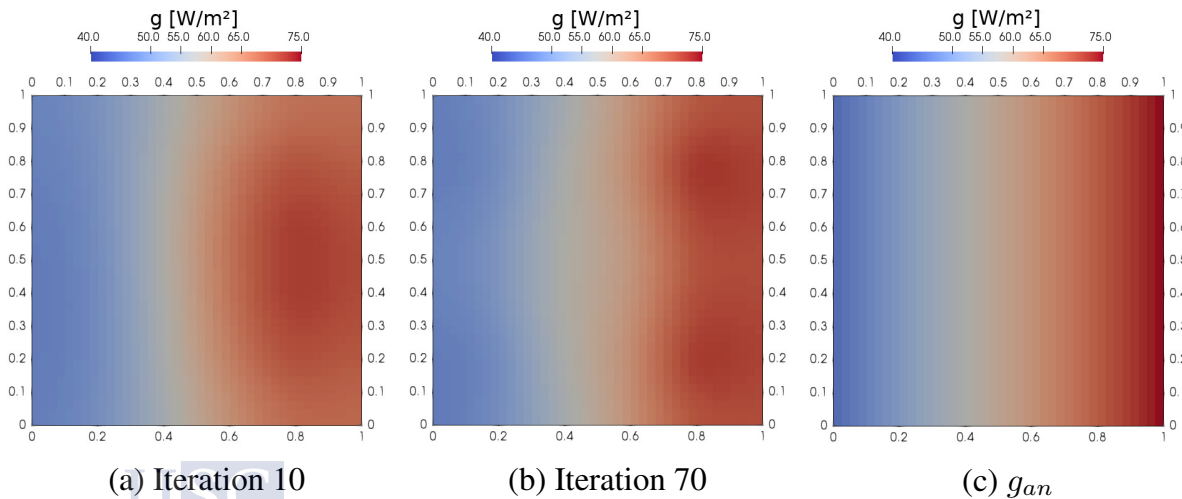


Figure 4.1.5: The estimated heat flux by Alifanov’s regularization at different iterations (a,b) is compared to the analytical value (c) in the Benchmark 1.

We now investigate how the grid refinement influences the performance of this

algorithm. Figure 4.1.6 shows the behavior of the relative error of the estimated heat flux with the grid refinement. This test is performed with the stopping criterion $J_1 < J_{1_{tol}} = 1e - 4K^2$. The error in general decreases by increasing the mesh refinement. However, the decrease is not monotonic with a small increase for the 40^3 elements grid.

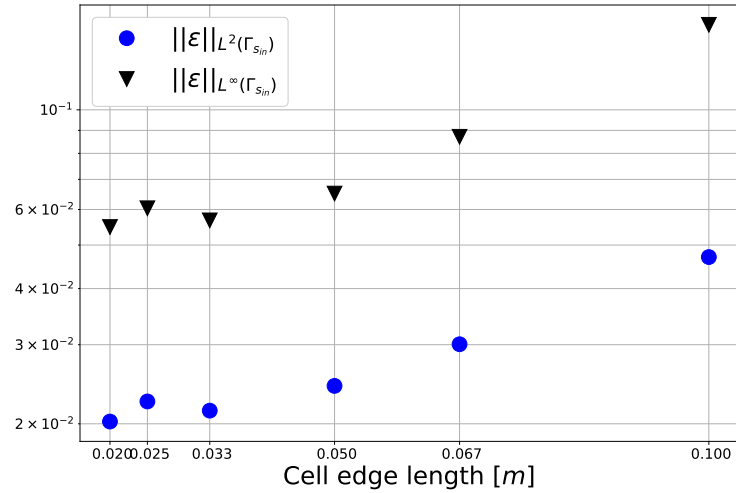


Figure 4.1.6: Behavior of the relative error norms (4.1.9) with the grid refinement in the Benchmark 1.

4.1.2.2 Parameterization of the Boundary Condition

We now test the performances of the parameterization method described in Section 3.1.2. In particular, we consider the effects that the basis functions parameters have on the results and the conditioning of the linear system (3.1.42). Moreover, also in this case, we test the effect of the mesh refinement on the estimated heat flux.

As already mentioned, we consider Gaussian RBFs as basis functions for the parameterization of the boundary heat flux. Recalling (3.1.24), the basis functions are given by

$$\phi_j(\mathbf{x}) = e^{-(\eta\|\mathbf{x}-\boldsymbol{\xi}_j\|_2)^2}, \quad \text{for } j = 1, 2, \dots, M,$$

where we locate the centers $\boldsymbol{\xi}_j$ at the projection of the virtual measurement points on the boundary Γ_{sin} according to (3.1.23).

Both the choice of the basis functions (3.1.24) and of the position of their center are arbitrary. However, they come suggested from the physics of the problem. The Gaussian RBFs are selected because their radial decay reduces the correlation between bases which are far away. For a similar reason, the RBFs are centered at the projection of the measurements to have a relationship between bases and measurements. This reasoning applies well to CC molds because we have the thermocouples located in a surface parallel and close to the boundary where we want to estimate the heat flux. In a more general scenario, these choices lose their motivation.

To completely define the basis functions, we still must tune the shape parameter η . Then, we first analyze the influence of η on the invertibility of system (3.1.42) and on the boundary heat flux estimation. This parameter controls the decay of the RBF. For bigger (smaller) values of η the decay is faster (slower). Figure 4.1.7 (a) shows the decay of the normalized singular values of $\Theta^T \Theta$ for different η . The singular values are normalized by dividing them all by the first one. In general, we can see that to bigger values of the shape parameter, correspond a slower decay of the singular values.

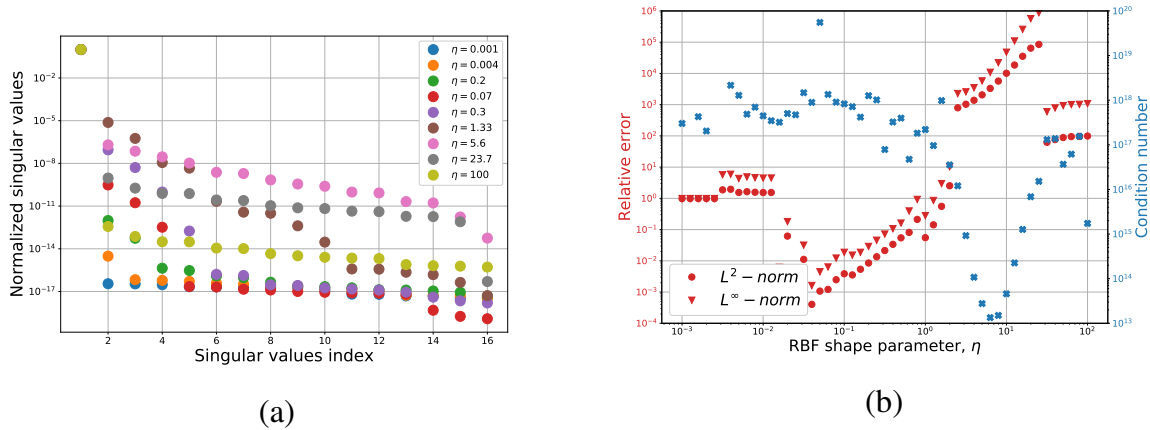


Figure 4.1.7: Effect of the RBFs shape parameter, η , on (a) the normalized singular values of the matrix $\Theta^T \Theta$ and on (b) the L^2 - and L^∞ -norms of the relative error and on the condition number of the linear system (3.1.42).

Figure 4.1.7 (b) shows the condition number of the linear system (3.1.42). The condition number is computed as the ratio between the bigger and the smaller singular value

$$\kappa_{\Theta^T \Theta} = \frac{\sigma_{max}}{\sigma_{min}}. \quad (4.1.10)$$

The figure shows it together with the L^2 - and L^∞ -norms of the relative error (4.1.9). The method used for the solution of the linear system (3.1.42) is LU factorization

with full pivoting. In the Figure 4.1.7 (b), we see that the best results are obtained for $\eta = 0.1$ (see Figure 4.1.8). Interestingly, looking at the behavior of the condition number, we notice that the quality of the results is not correlated to the conditioning of (4.1.9).

To have qualitatively insight on the results, Figure 4.1.8 show the true and the computed heat flux together with the relative error between the two. From this figure, we appreciate the outstanding performance of this method.

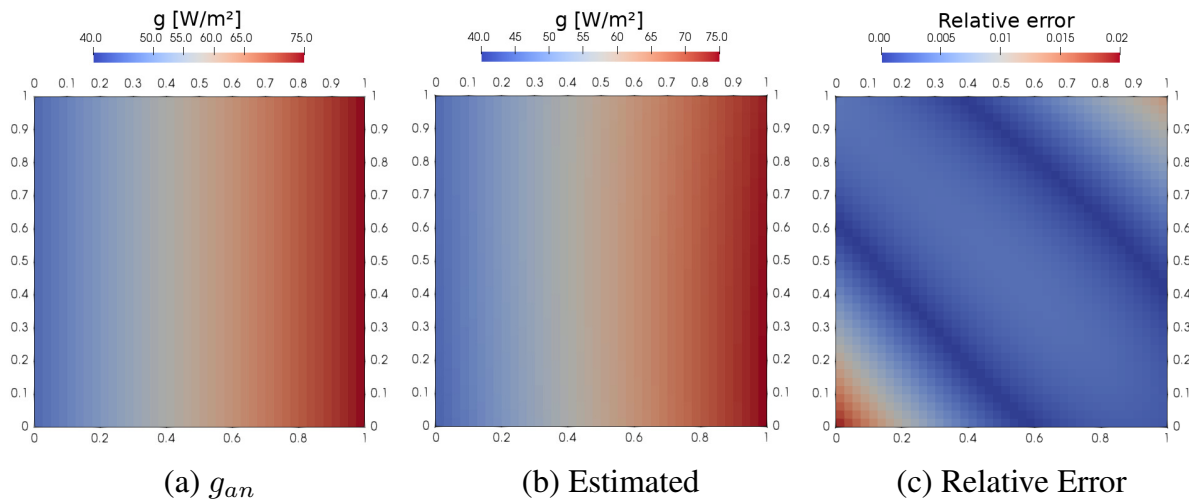


Figure 4.1.8: Comparison of the analytical (a) and estimated (b) boundary heat flux for the Benchmark 1 together with the relative error between the two (c). This result is obtained by using the parameterization method with RBF shape parameter $\eta = 0.1$.

As for Alifanov’s regularization, we test the effects of grid refinement on the estimation of the heat flux. Figure 4.1.9 shows that this method is almost insensitive to the mesh refinement. In fact, the error oscillates between two very close values.

4.1.2.3 Effects of Measurements Noise

In all previous tests, we considered the measurements to be free of noise. This is not the real case. In fact, thermocouples measurements are notoriously noisy. Thus, we analyze in this section the effects that the measurement noise has on the algorithms performances. From the industrial point of view, this analysis is of particular interest for our application.

We perform this analysis by adding to the measurements vector the Gaussian

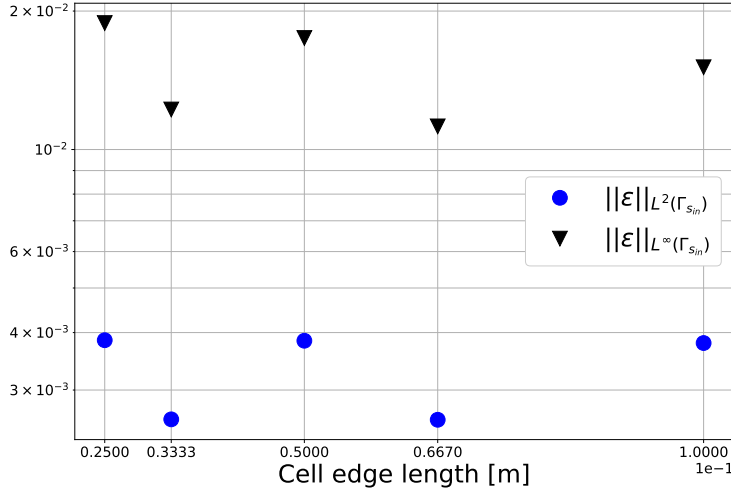


Figure 4.1.9: Behavior of the relative error norms (4.1.9) as a function of the grid refinement using the parameterization method for Benchmark 1.


random noise \mathbf{w}_n

$$\hat{\mathbf{T}}_{\mathbf{w}} = \hat{\mathbf{T}} + \mathbf{w}_n, \quad \mathbf{w}_n = \mathcal{N}(\boldsymbol{\mu}, \Sigma), \quad (4.1.11)$$

where, $\boldsymbol{\mu} \in \mathbb{R}^M$ is the mean vector and $\Sigma \in \mathbb{M}^{M \times M}$ is the covariance matrix. In particular, we choose \mathbf{w}_n to be an Independent and Identically Distributed (IID) random variable with zero mean, i.e. $\mathbf{w}_n = \mathcal{N}(\mathbf{0}, \omega^2 I)$.

To study the effect of noise, we perform several solutions of the inverse problem using $\hat{\mathbf{T}}_{\mathbf{w}}$ as thermocouples’ measurements. For each test, we compute 200 samples. All these computations are done on the 40^3 elements grid. Then, we analyze the statistical and qualitative properties of the obtained results. In our first test, we analyze the behavior of the relative error (4.1.9) for different values of the noise standard deviation ω .

Using Alifanov’s regularization for the minimization of J_1 , we must use a stopping criterion that regularize the solution. In fact, the regularization parameter is the iteration counter i . Here, we use the Discrepancy Principle (DP) as stopping criterion (see e.g [7]). Thus, the iterations are stopped when



$$J_1[g^{i+1}] < \left(\frac{\omega^2 M}{2} \right)^2, \quad (4.1.12)$$

where M is the number of thermocouples.

Figure 4.1.10 illustrates the results of this first test. We notice that Alifanov’s regularization is able to filter the noise only for $\omega < 0.02$. On the other hand, we

see that for the parameterization method with LU factorization the results are spread around the mean value. It suggests that the noise is propagating from the measurements into the solution. As anticipated at the end of Section 3.1.2, we require some regularization technique in solving (3.1.42) to make it robust with respect to noise in the measurements.

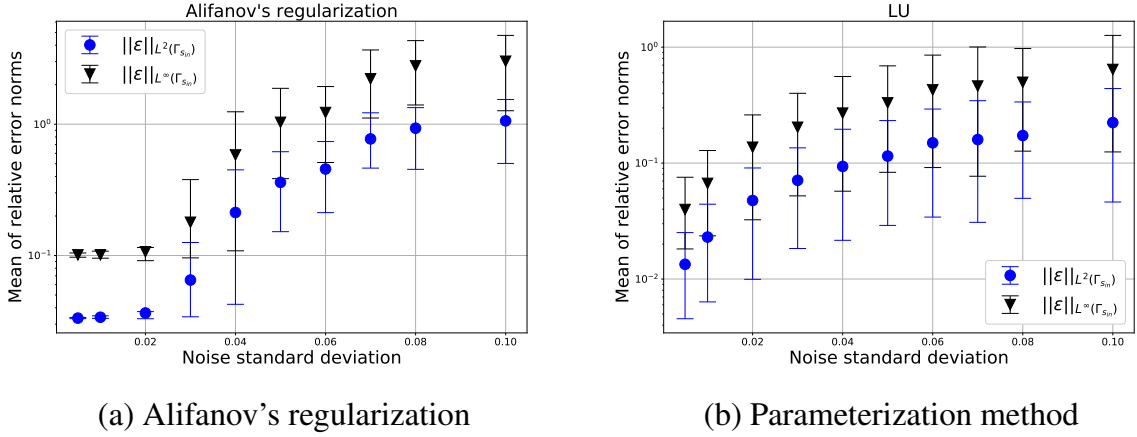


Figure 4.1.10: Behavior of the relative error L^2 - and L^∞ -norm as a function of the standard deviation of the noise in the measurements for Alifanov's regularization (a) and the parameterization method (b) in Benchmark 1 (90% quantile bars shown).

As described in Section 3.1.2, we use TSVD regularization in the parameterization method. We opt for this technique because it is effective when we have jumps in the singular values decay (see Figure 4.1.7). As already said, attention must be paid when using regularization techniques in selecting the regularization parameter. In our case, the regularization parameter, α_{TSVD} is the number of singular values used in the truncation. Different methodologies are available in the literature, e.g. Unbiased Predictive Risk Estimator (UPRE), DP, L-curve, U-curve, Generalized Cross Validation (GCV) (see [7]). However to show the dependency of the results on the regularization parameter, we performed numerical tests.

Figure 4.1.11 shows the behavior of the L^2 - and L^∞ -norm of the relative error with respect to regularization parameter α_{TSVD} , for different values of the noise standard deviation ω . As expected, the optimal value of the regularization parameter depends on the noise variance. In fact, for low noise level we should use higher values of α_{TSVD} reducing it as the noise increases and vice versa.

Testing again the TSVD regularization fixing α_{TSVD} and increasing the noise standard deviations, we clearly see the regularizing effect of the TSVD. Figure 4.1.12 shows the obtained results. In the figure, we appreciate the importance of a right

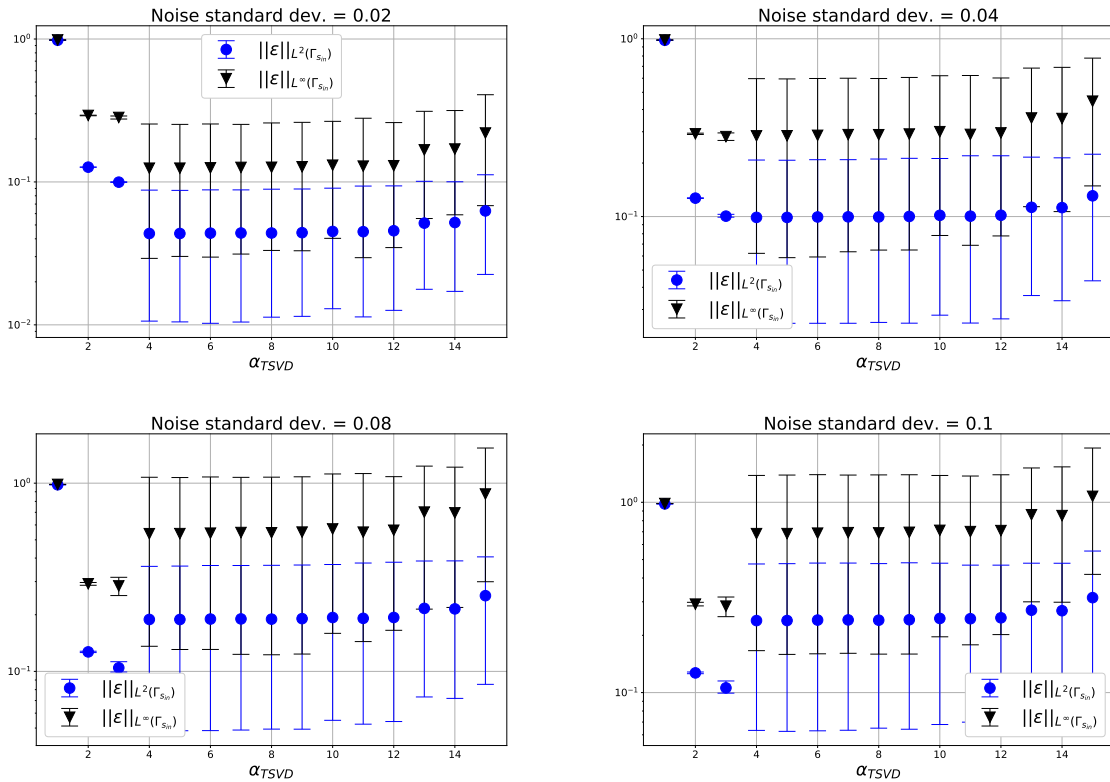


Figure 4.1.11: Effect of the regularization parameter α_{TSVD} using the TSVD in parameterizing the heat flux to solve the inverse problem of Benchmark 1 (90% quantile bars).

choice of the regularizing parameter. In fact, if we have very low noise in the measurements, we should opt for higher values of α_{TSVD} and vice versa.

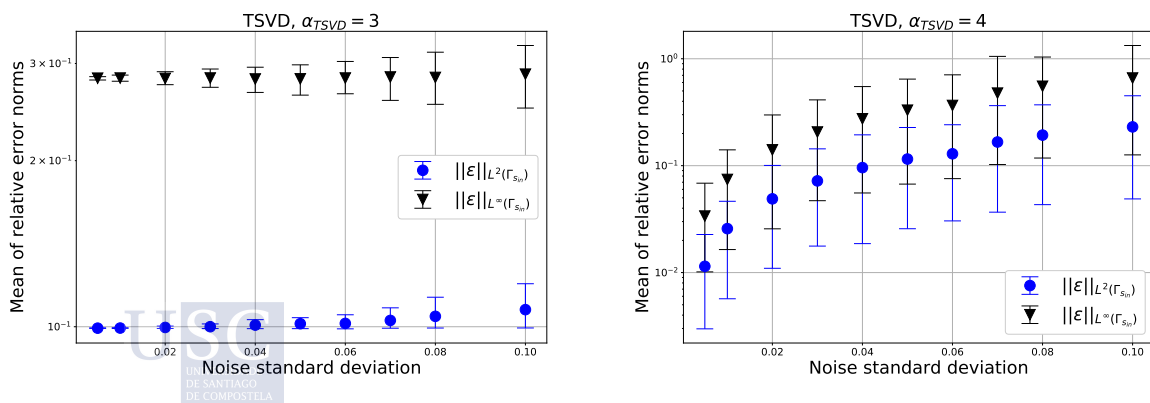


Figure 4.1.12: Behavior of the relative error with respect to the standard deviation of the noise in the measurements using the parameterization method with TSVD regularization in Benchmark 1 (90% quantile bars shown).

We conclude this noise analysis by looking at a realization of the computed heat flux with the different methods. Figure 4.1.13 provides a qualitative example of the performances of the inverse solvers for $\omega = 0.08$. As expected, the noise is not well filtered by the Alifanov's regularization while the parameterized method with TSVD provides a smooth solution in good agreement with the true value.

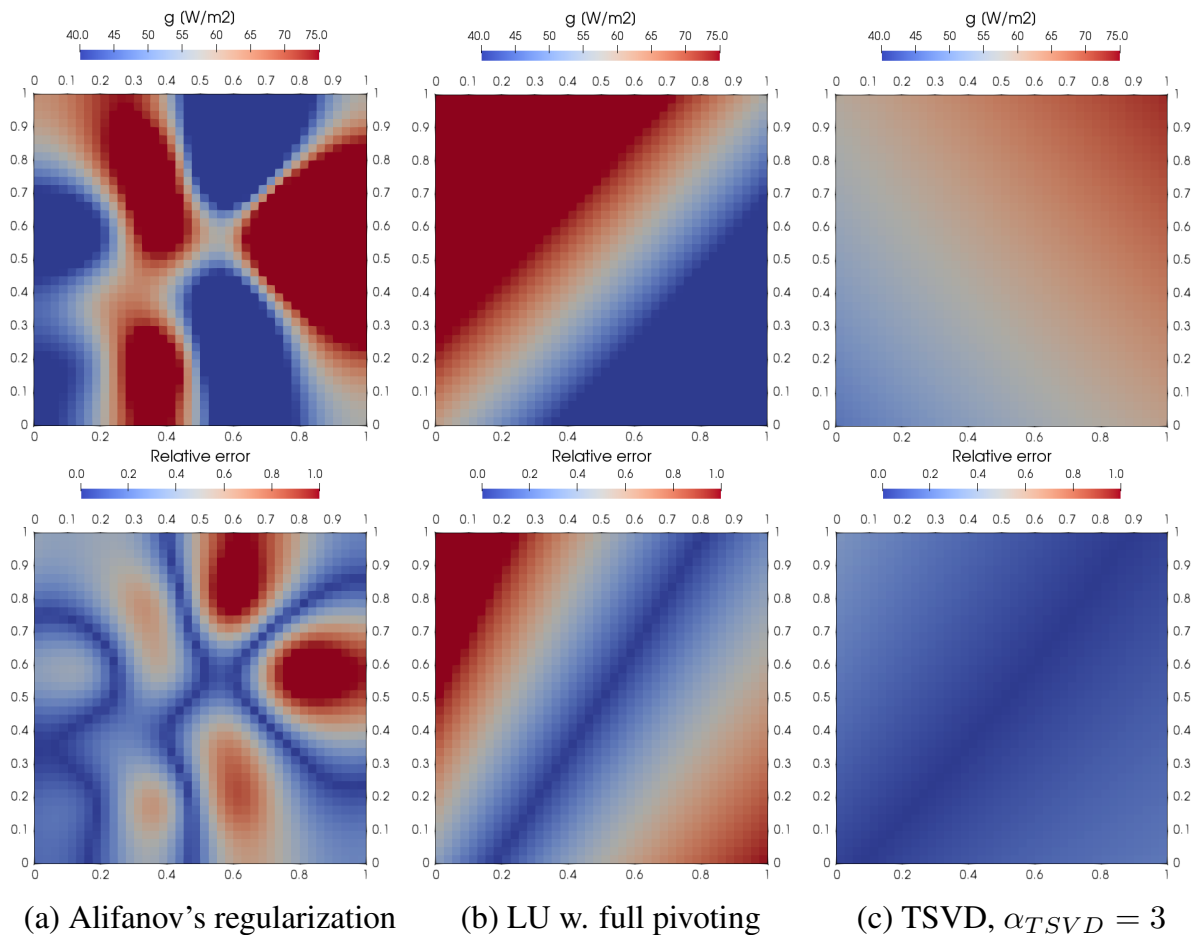


Figure 4.1.13: Comparison of the estimated heat flux for Alifanov's regularization (a) and parameterization method with (c) and without (b) regularization for the analytical benchmark case with noisy measurements (noise standard deviation $\omega = 0.08$).

4.1.2.4 Effects of Changing Measurement Points Positioning and Number

In this section, we test the influence of the thermocouples positioning and number on the estimation of the heat flux. Firstly, we analyze the effects of changing the location of the thermocouples on the y -axis to investigate the importance of the thermocouples distance from the boundary where we are estimating the heat flux. Secondly, we study the behaviour of the proposed inverse solvers as we change the number of measurement points in the thermocouples plane.

Figure 4.1.14 shows the values of the L^2 - and L^∞ - norms of the relative error (4.1.9) as we move the thermocouples plane away from Γ_{sin} . We provide the results for both inverse solvers.

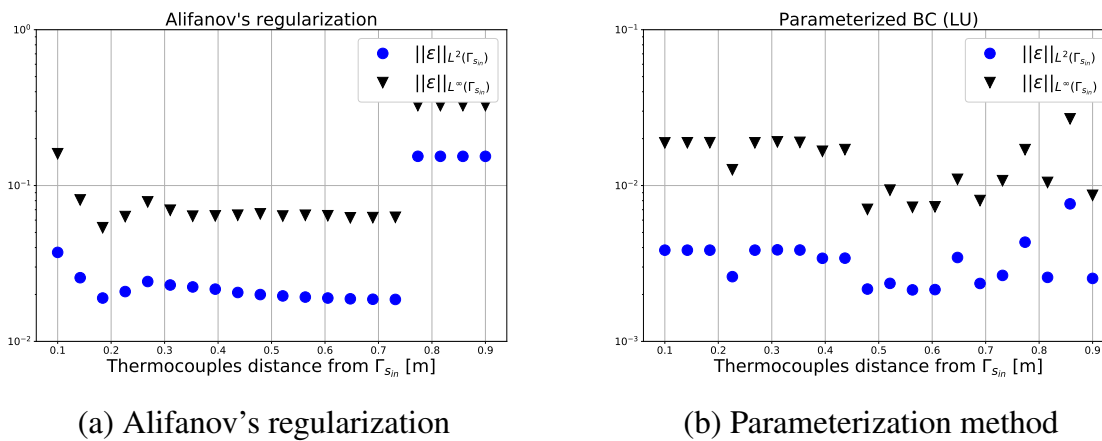


Figure 4.1.14: Behaviour of the relative error when changing the distance of the thermocouples from the boundary Γ_{sin} where the heat flux g is estimated.

The error norms show an interesting behaviour. For Alifanov's regularization, they have a fast decay until the thermocouples plane is 0.3 m away from Γ_{sin} . Then after a small increase, the error reaches a plateau before jumping to higher values when the thermocouples plane has y -coordinate higher than 0.7 m. On the other hand, the parameterization method shows a different behaviour in Figure 4.1.14 (b). This method shows little dependency on the thermocouples distance from the boundary, providing similar solutions for the different thermocouples positions.

Now, we analyze the influence of the thermocouples number. We design the following test: fixed the thermocouples y -coordinate, we change the number of thermocouples on that plane. The thermocouples are located in a uniform lattice with equal number of thermocouples on the x - and z -axis. Figure 4.1.15 shows the obtained

results for this test using both inverse solvers.

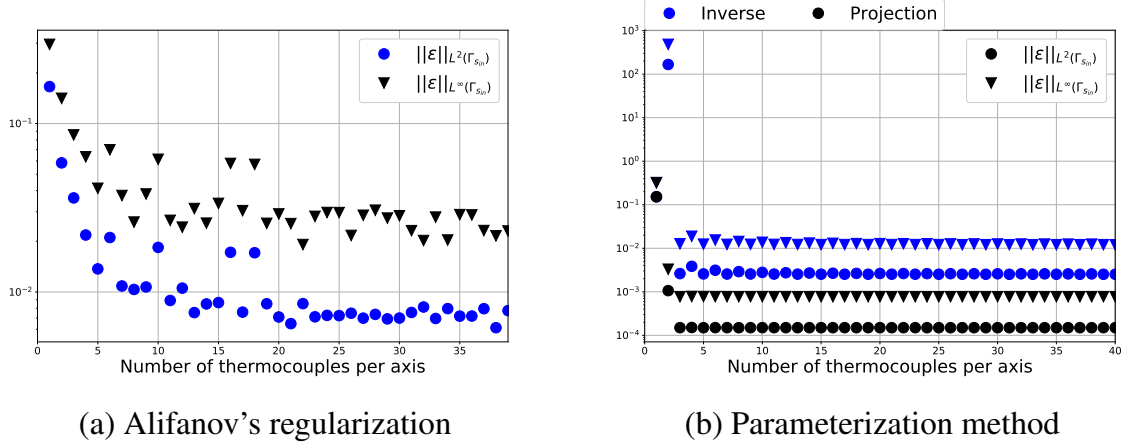


Figure 4.1.15: Behaviour of the relative error when changing the number of thermocouples on the plane $y = 0.2 m$. On the x -axis of the graph, we have the number of thermocouples on each axis of the lattice. Then, the total number of thermocouples is the square of the shown value. In the figure (b), the blue results are obtained by solving the inverse problem and the black ones are the best possible approximation of the true heat flux in the parameterized space (remember that by increasing the number of thermocouples we increase the number of basis of the heat flux parameterization).

Also in this test, the behaviour of the two methods is very different. For Alifanov's regularization, we have a fast decay of the error before reaching a plateau. On the other hand, the parameterization method has very little dependence on the number of thermocouples for more than 9 thermocouples.

4.1.3 Inverse Problem with Temperature and Total Heat Flux Measurements

In this section, we discuss the numerical solution of the inverse Problem 3.6 where $T_s[g](\mathbf{x}_i)$ is the solution of Problem 4.1 at points \mathbf{x}_i , for all $i = 1, 2, \dots, M$ and $\hat{G} = \int_{\Gamma_{sin}} g_{an} d\Gamma$, g_{an} being defined by (4.1.7). All computations are performed on the 40^3 elements grid and the basis in the parameterization method are as in the previous section.

With respect to the previous section, we have one additional parameter: the total heat weight, p_g . Since, it is not possible to set it a priori, we analyze its effects on the solution. Figure 4.1.16 (a) shows the behavior of the L^2 - and L^∞ -norm of the relative error for different values of p_g using Alifanov's regularization for the solution

of the inverse problem. On the other hand, Figure 4.1.16 (b) shows the same graph for the parameterization method with LU decomposition with full pivoting. These computations are performed without errors in the measurements.

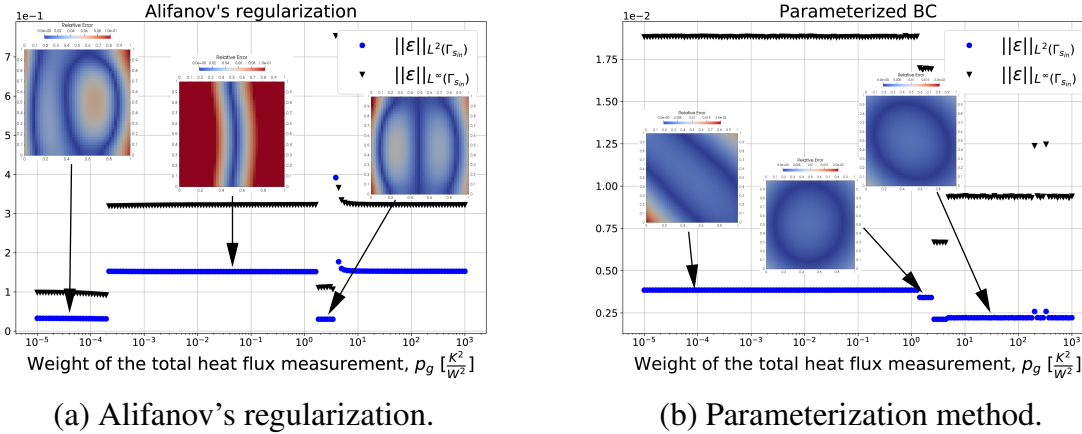


Figure 4.1.16: Behavior of the relative error with respect to the total heat measurement weight, p_g , using Alifanov's regularization (a) and parameterization of the heat flux with LU decomposition with full pivoting (b) in the analytical benchmark case.

Comparing the two figures (notice the different order of magnitude on the y-axis), we see that adding the total heat measurement improves the boundary heat flux estimation only for the parameterization method. In Alifanov's regularization, we have a very small decrease of the relative error for $p_g \lesssim 1e - 4 \frac{K^2}{W^2}$. For higher values, the error jumps to a constant higher value. Moreover, in Figure 4.1.16 (a), we appreciate an interesting jump in the error for $1.5 \frac{K^2}{W^2} < p_g < 3 \frac{K^2}{W^2}$. For these values, we recast similar results to those obtained for $p_g \lesssim 1e - 4 \frac{K^2}{W^2}$.

Figure 4.1.17 provides further information on the effect of p_g . In the parameterization method, p_g does not have any effect on the solution for $p_g < 1$. Then for higher values of p_g , the relative error decreases linearly. On the other hand, the figure confirms the interesting behavior of Alifanov's regularization for $1.5 \frac{K^2}{W^2} < p_g < 4 \frac{K^2}{W^2}$. However, it shows also for this method an almost linear decrease of the total heat relative error for $p_g > 4 \frac{K^2}{W^2}$.

We now analyze the influence that the noise in the measurements has on the estimation of the heat flux, g . We perform the following test: we select a noise level for the thermocouples measurements and test the effect of different noise levels on the total heat measurement, G . Figure 4.1.18 summarizes the results for Alifanov's regularization and parameterization method with full pivoting LU.

Notice that as expected for Alifanov's regularization, we have an increase of the

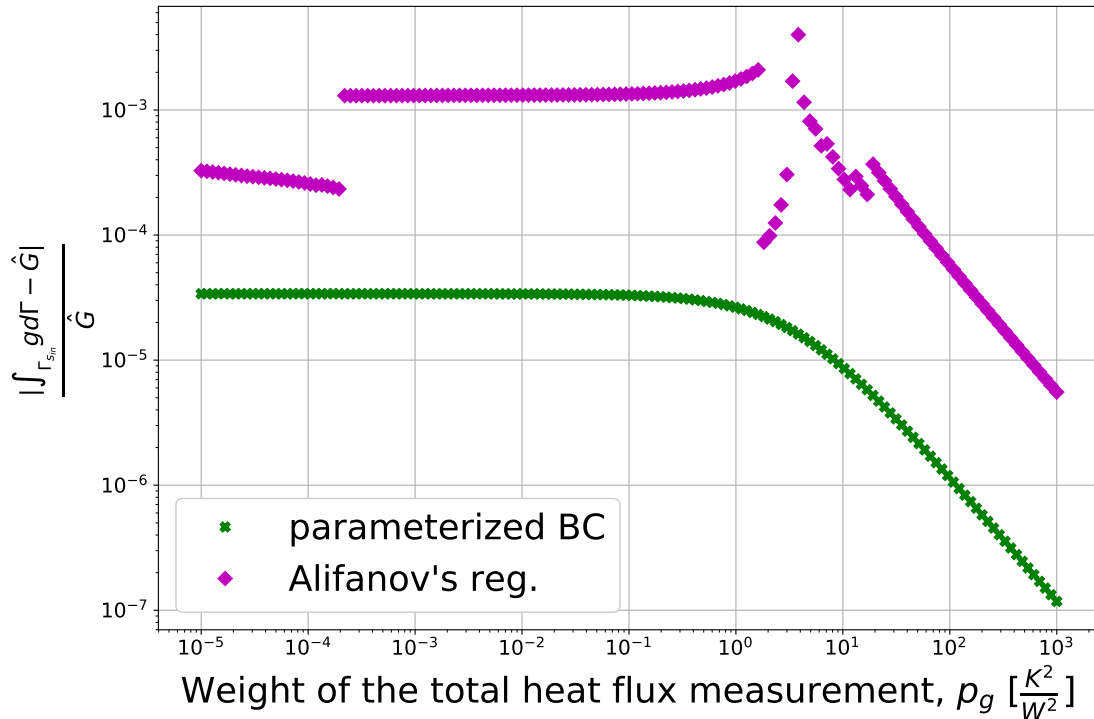


Figure 4.1.17: Behaviour of the relative error in computing the total heat flux on $\Gamma_{s_{in}}$ with respect to p_g .

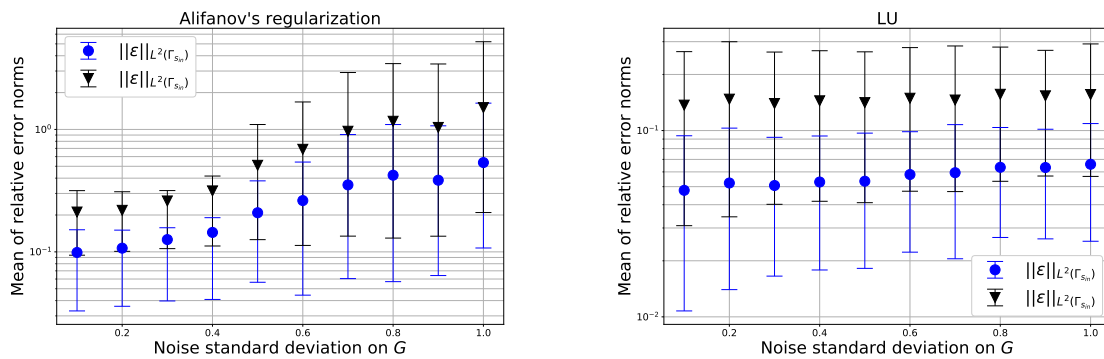


Figure 4.1.18: Effect of noise in the total heat flux measurement with 2% standard deviation noise in the thermocouples (90% quantile bars).

error increasing the noise on G . On the other hand, the parameterization method is not much influenced by this noise. However as in the previous section, we require some regularization for this method to filter out the measurement noise.

Also in this setting, we use TSVD for the regularization of (3.2.16). Figure 4.1.11

shows the behaviour of the L^2 - and L^∞ -norm of the relative error with respect to regularization parameter α_{TSVD} , for $\omega = 0.04$ and a 10% standard deviation noise on the total heat flux measurement, G . We can see that also in this case, with $\alpha_{TSVD} = 3$ TSVD is able to reduce the error both in the mean and in the standard deviation. For a lower number of singular values, we get a very smooth solution but too regularized while for $\alpha_{TSVD} > 3$, we do not filter out the noise.

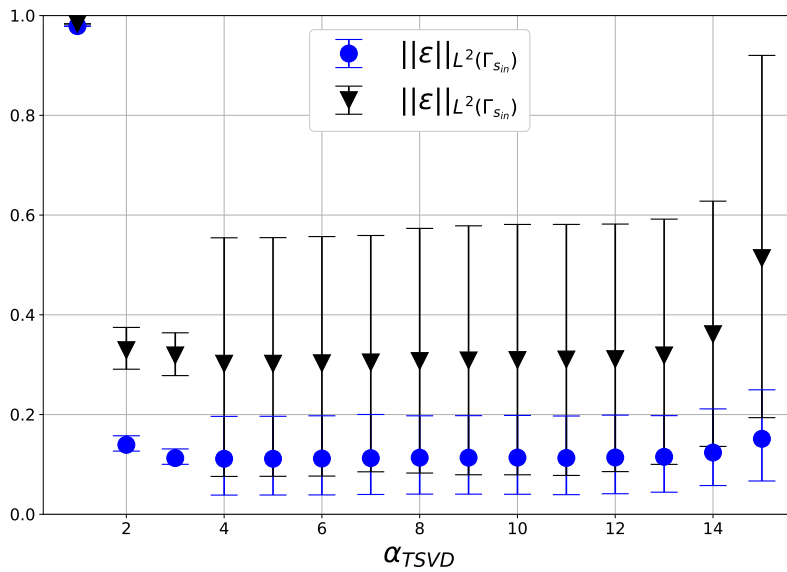


Figure 4.1.19: Effect of the regularization parameter α_{TSVD} using the TSVD in parameterizing the heat flux to solve the inverse problem of Benchmark 1 with 90% quantile (noise standard deviation 0.04 in the thermocouples and 0.1 in the total heat measurement).

4.1.4 Conclusions

To draw conclusions on the performances of the tested inverse solvers, we compare their computational cost. This is of particular interest in our research because we want to achieve real-time performances. Table 4.3 illustrates the Central Processing Unit (CPU) time required for the computations with no error in the measurements and $J_{tol} = 10^{-4}$ in the case of only temperature measurements available. Notice that all the computations were performed in serial on a Intel[®] Core[™] i7-8550U CPU processor.

These results confirm that the offline-online decomposition makes the parameterized heat flux method eligible for real time applications. On the other hand,

Table 4.3: Inverse problem CPU time comparison for Benchmark 1.

	Alifanov's reg.	Parameterized heat flux	
		offline	online
CPU time	18.8 s	7.21 s	0.0056 s

Alifanov's regularization requires several solutions of direct, adjoint and sensitivity problems, so it cannot be employed in real time as it is.

With this final remark, we conclude that the parameterization method outclasses Alifanov's regularization both in the quality of the estimation provided and in the robustness (with TSVD regularization) with respect to errors in the measurements.

Moreover, thanks to its offline-online decomposition, the parameterization method has proved to be able to achieve real-time computation. In fact, it requires a computationally expensive offline computation in which we solve several direct problems. In the online phase it is very fast since we only solve a linear system with dimension equal to the number of basis used in the parameterization of the heat flux.

Finally, we considered the case of having as data for the inverse problem also a total heat flux measurement. The parameterization method results are improved under every aspect by introducing this additional data. On the other hand, Alifanov's regularization is only slightly affected by this additional data.

4.2 BENCHMARK 2

The benchmark case presented in this section is a numerical test case. This benchmark is designed to mimic the real industrial scenario of a CC mold. In particular, the domain is a simplification of a mold plate and the physical quantities have typical industrial values. Also the thermocouples number and positioning are those of a real mold.

In this benchmark, the direct problem does not have an analytical solution. Then, we assume that the direct problem is well solved and focus our attention on the solution of the inverse problem. To test the inverse solvers, we choose a priori a mold-steel heat flux, g_{true} . By solving the direct problem for this heat flux, we obtain the related temperature field. Then, we use its values at the thermocouples points as measurements to input to the inverse solver. Table 4.4 summarizes the physical properties for this test case and the chosen heat flux, g_{true} .

As for the previous benchmark, the direct problem is a steady-state heat conduction problem in an homogeneous isotropic solid with a rectangular parallelepiped domain. The domain Ω_s is as in Figure 4.1.1 with $\Gamma_{sex} = \Gamma_I \cup \Gamma_{II} \cup \Gamma_{III} \cup \Gamma_{IV}$. The mathematical formulation of the direct problem is that of Problem 2.6.

Table 4.4: Parameters used for the simulation of Benchmark 2.

Parameter	Value
Thermal conductivity, k_s	300.0 $W/(mK)$
Heat transfer coefficient, h	5.66e4 $W/(m^2K)$
Water temperature, T_f	303 + 8 · (1.2 - z) K
Heat flux, g_{true}	$2e5(x - 1)^2 - 2e5 \cdot z - 5e5$ W/m^2
L	2 m
W	0.1 m
H	1.2 m

For the discretization of the domain, we use a structured orthogonal grid with uniformly distributed elements along the three axes. We use 200, 50 and 100 elements on the x -, y - and z -axis respectively. Thus, the grid size is $1e6$ elements.

As in the real industrial case under study, we locate the virtual thermocouples in the plane $y = 0.02$ m . In this plane, they are equally distributed on the x - and z -axis as shown in Figure 4.2.1.

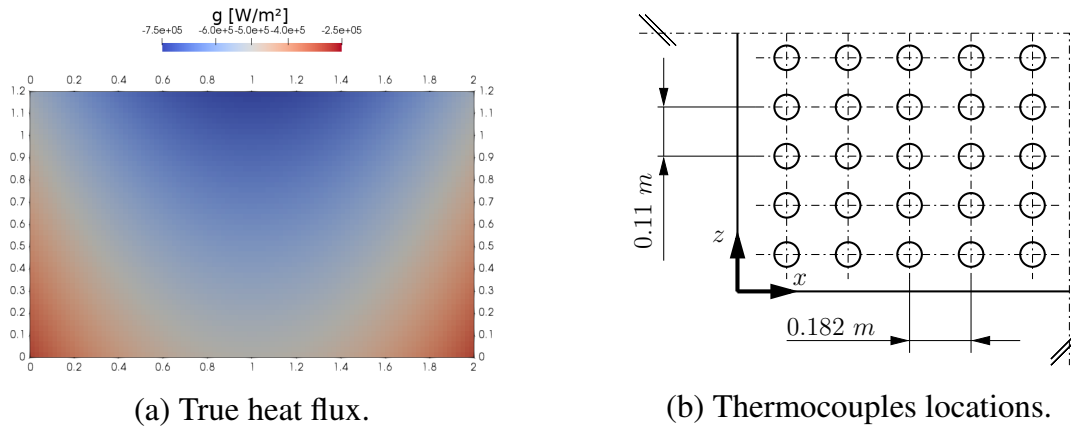


Figure 4.2.1: True heat flux (a) and position of the 100 thermocouples at the plane $y = 0.02$ m (b) for Benchmark 2.

4.2.1 Inverse Problem with Temperature Measurements

In the present section, we analyze the performance of the proposed methods in the solution of the inverse Problem 3.1 for the presented numerical test case. First, we analyze the performances of Alifanov’s regularization (see Section 3.1.1). Table 4.5 shows the parameters used for the simulation.

Figure 4.2.2 illustrates the estimated heat flux, g , at different iterations of the

Table 4.5: Parameters used in the Alifanov's regularization algorithm.

Parameter	Value
g^0	$0 \frac{W}{m^2}$
J_{tol}	$1e2 K^2$
$\frac{\ J^n - J^{n-1}\ }{J^n}$	$1e - 2$

algorithm. We notice that the algorithm provides a solution not in agreement with g_{true} . In particular, it overestimates the heat flux close to the measurement points while far from the measurements the initial estimate is not modified. Moreover, increasing the number of iterations do not improve the results. Due to the inability of estimating the heat flux also in the simplest case without measurement noise, we do not perform further tests with this method.

We now consider the parameterization method of Section 3.1.2. As for the previous benchmark, we start by performing a numerical analysis on the influence of the RBF shape parameter, η , on the invertibility of system (3.1.42) and on the estimated heat flux. Figure 4.2.3 (a) shows the decay of the singular values of $\Theta^T \Theta$ for different η . In general, we can see that to bigger values of the shape parameter, correspond a slower decay of the singular values. Moreover, we see from this singular value decay and in Figure 4.2.3 (b) that for $\eta > 1$ the condition number of the system decreases. However, the relative error in the heat flux estimation increases significantly for these values of η .

To conclude, there is no relationship between the condition number of the linear system and the obtained results for this test. However according to Figure 4.2.3 (b), we obtain the best results for $\eta = 0.3$. Figure 4.2.4 shows the results obtained for this value of the RBF shape parameter. Then, we use this value in the following tests.

We now analyze the effect of noise in the measurement. Figure 4.2.5 shows the effect of different noise levels on L^2 - and L^∞ -norms of the relative error (4.1.9) using LU factorization with full pivoting for the solution of (3.1.42). The relative error increases linearly with the noise level.

As for the previous benchmark, we test the regularization properties of TSVD on this problem. Figure 4.2.6 shows the effect of the regularization parameter α_{TSVD} on the L^2 - and L^∞ -norms of the relative error for different values of the noise standard deviation, ω . As expected, the optimal value of the regularizing parameter α_{TSVD} decreases as the noise increases. However, for all the considered cases, we are able to achieve a relative error that in the L^2 -norm is below 2%.

To conclude, Figure 4.2.7 shows the behavior of the relative error increasing the measurement noise for $\alpha_{TSVD} = 5$ and $\alpha_{TSVD} = 7$. Notice that also for severe

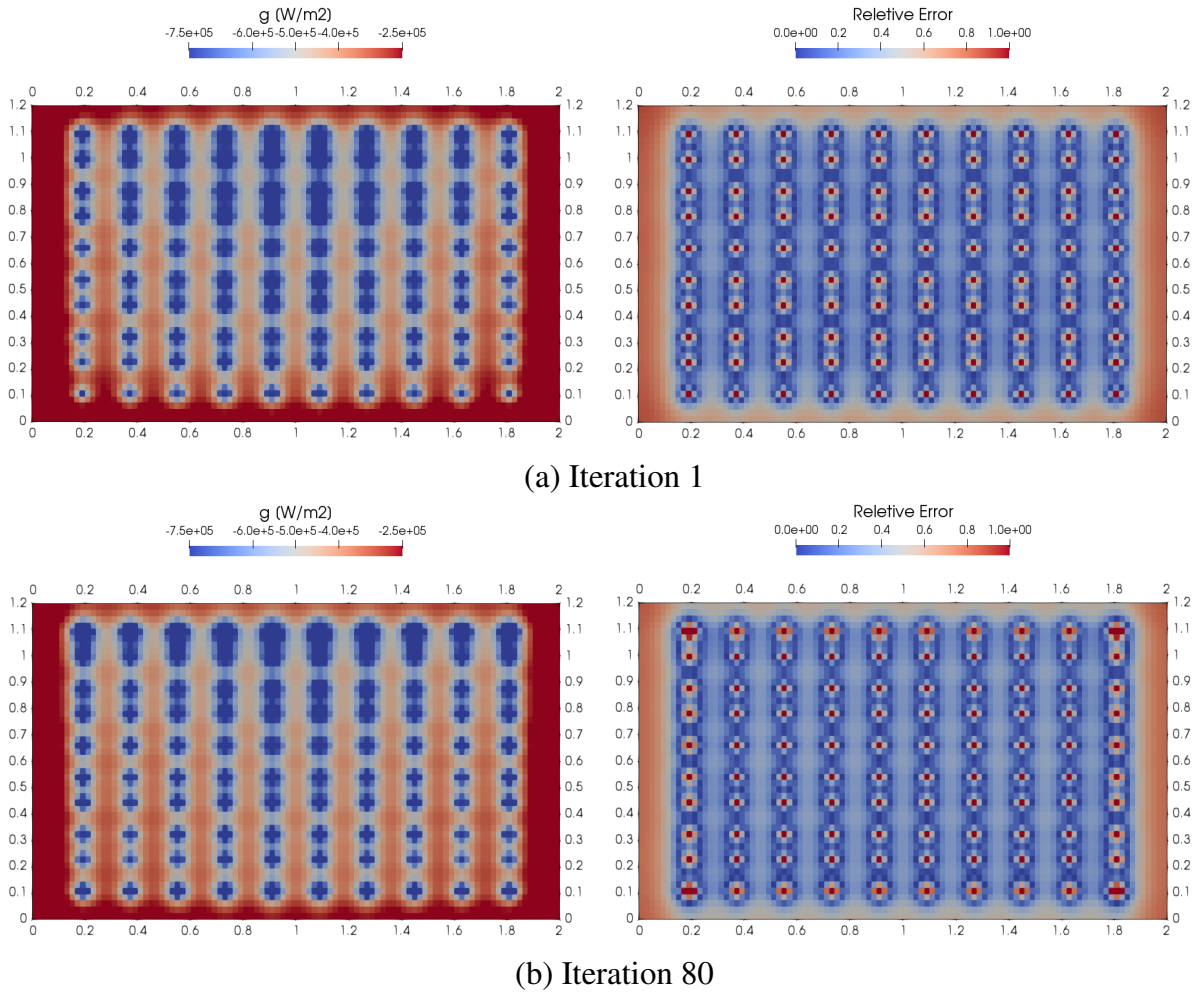


Figure 4.2.2: Comparison of the computed heat flux by Alifanov's regularization at different iterations.

noise in the thermocouples measurements, we are able to obtain a valid reconstruction of the boundary heat flux.

4.2.2 Inverse Problem with Temperature and Total Heat Flux Measurements

In this section, we discuss the numerical solution of the inverse Problem 3.6 where $T_s[g](\mathbf{x}_i)$ is the solution of Problem 2.6 at points \mathbf{x}_i , for all $i = 1, 2, \dots, M$, g_{true} as in Table 4.4 and $\hat{G} = \int_{\Gamma_{sin}} g_{true} d\Gamma$.

With respect to the previous section, we have one additional parameter: the total heat weight, p_g . Since, it is not possible to set it a priori, we analyze its effects

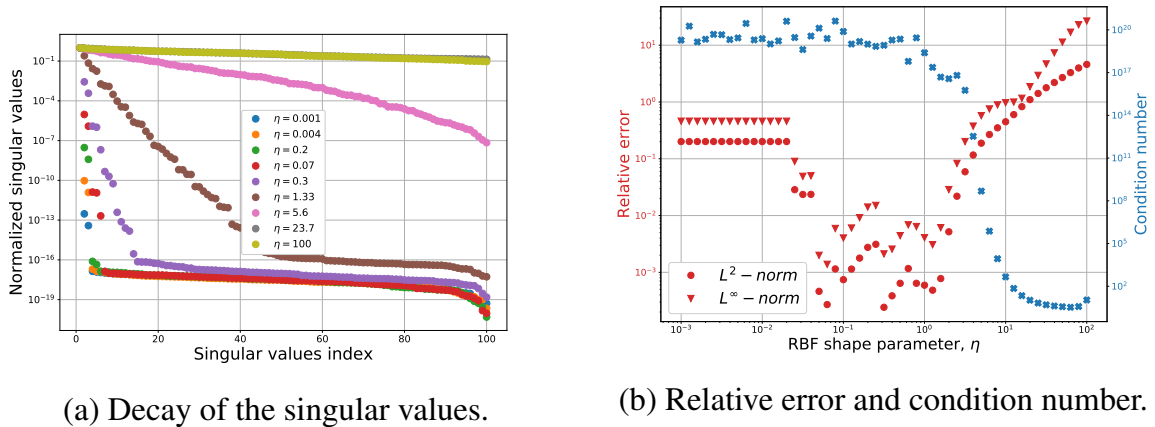


Figure 4.2.3: Effect of the RBFs shape parameter on (a) the singular values of the matrix $\Theta^T \Theta$ and on (b) the L^2 - and L^∞ -norms of the relative error (4.1.9) using LU with full pivoting and on the condition number of (3.1.42).

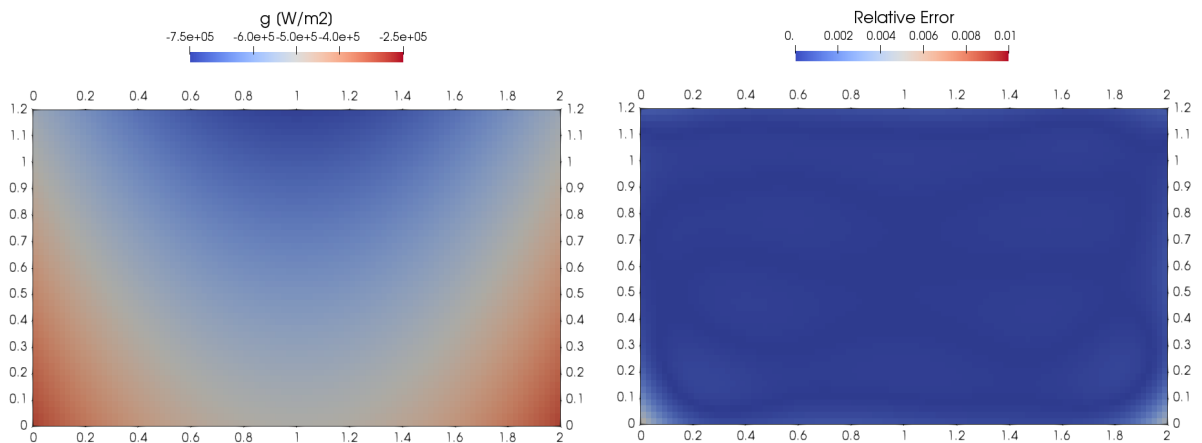


Figure 4.2.4: Estimated heat flux (a) and the respective relative error (b) using the parameterization method with RBF shape parameter $\eta = 0.3$ in the industrial benchmark case.

on the solution. Figure 4.2.8 shows the behavior of the L^2 - and L^∞ -norm of the relative error for different values of p_g using Alifanov’s regularization (a) and the parameterization method with LU factorization (b) for the solution of the inverse problem. All these computations are performed without noise in the measurements.

Analyzing Figure 4.2.8, we appreciate a different behavior for the two methods. Alifanov’s regularization improves its results, reaching a minimum of the relative error for $p_g \approx 1e-8$. Then, the error goes quickly to a plateau in which the estimated heat flux is uniform. On the other hand, the parameterization method error increases at jumps with p_g .

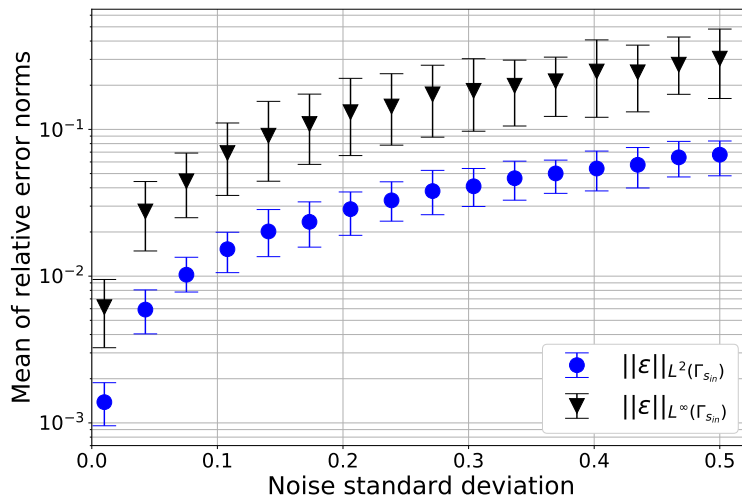


Figure 4.2.5: Effect of the measurements noise on the solution of the parameterization method with LU factorization with full pivoting in Benchmark 2 (90% quantile bars shown).

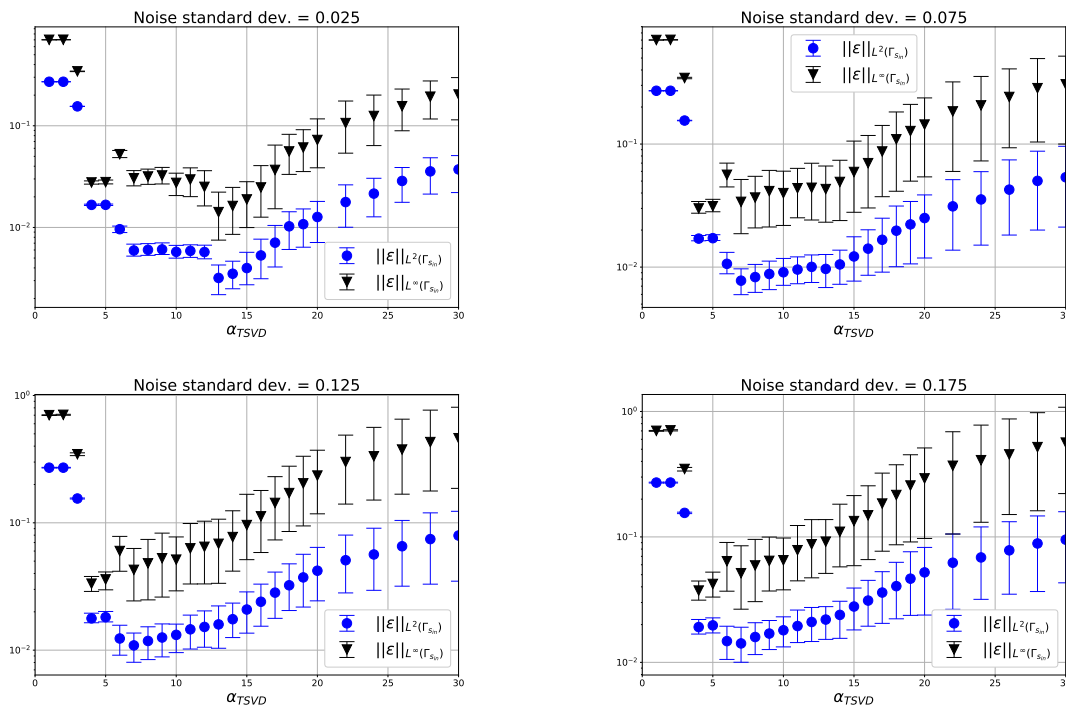


Figure 4.2.6: Effect of the regularization parameter α_{TSVD} using the TSVD in parameterization method for the industrial benchmark case (90% quantile bars shown).

Figure 4.2.9 shows the relative error on the total heat flux. It also provides in-

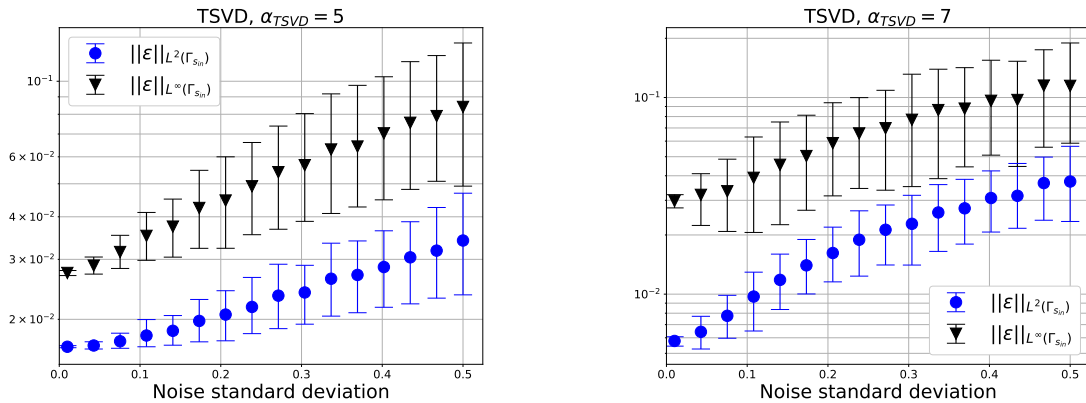
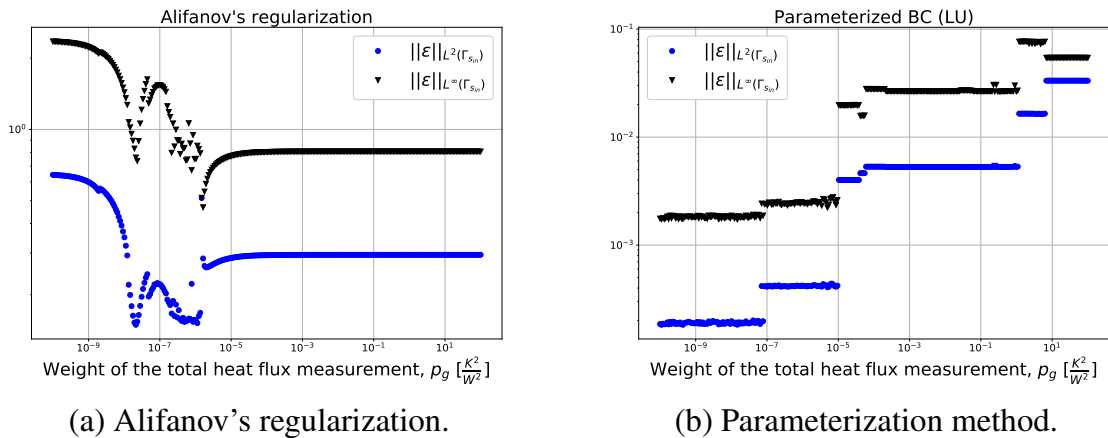


Figure 4.2.7: Behavior of the relative error with respect to the standard deviation of the noise in the measurements using the parameterization method with TSVD regularization in the industrial benchmark case (90% quantile bars shown).



(a) Alifanov's regularization.

(b) Parameterization method.

Figure 4.2.8: Effect of the functional weight p_g on the L^2 - and L^∞ -norms of the relative error (4.1.9) for the Alifanov's regularization (a) and the parameterization method with LU factorization (b). The thermocouples' measurements are free of noise.

interesting information. While both methods linearly improve their performance for $p_g > 1e - 5$, the parameterization method shows a very peculiar dependence on the weight for lower values. However, the parameterization method has a relative error two orders of magnitude smaller with respect to Alifanov's regularization.

We now test the effect of noise in the total heat measure \hat{G} for the parameterization method. Figure 4.2.10 shows the results obtained by fixing a the noise on the thermocouples and varying the noise on \hat{G} . As for Benchmark 1, the noise on \hat{G} do not affect significantly the estimated heat flux, g . Consequently, Figure 4.2.11 shows

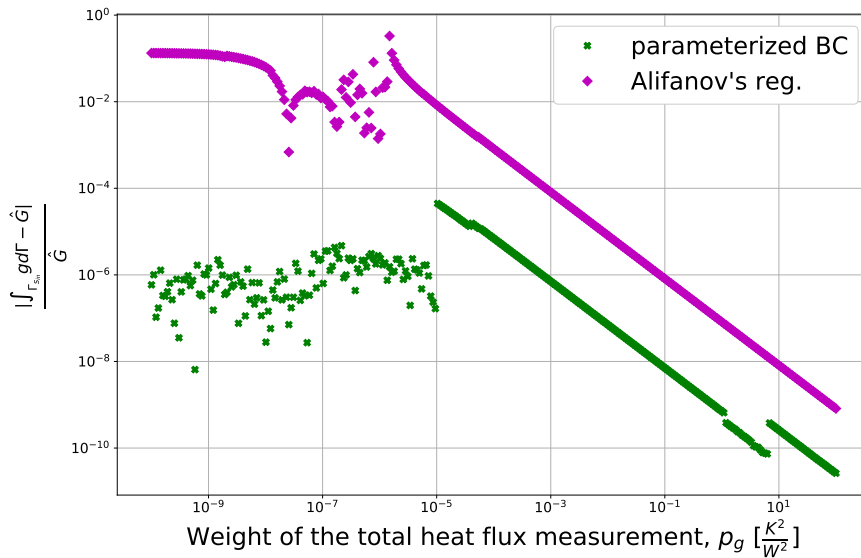


Figure 4.2.9: Effect on the total heat measurement weight p_g on the relative error in the total heat.

that the TSVD performs similarly to Section 4.2.1 with the optimum value for the regularization parameter being $\alpha_{TSVD} = 7$.

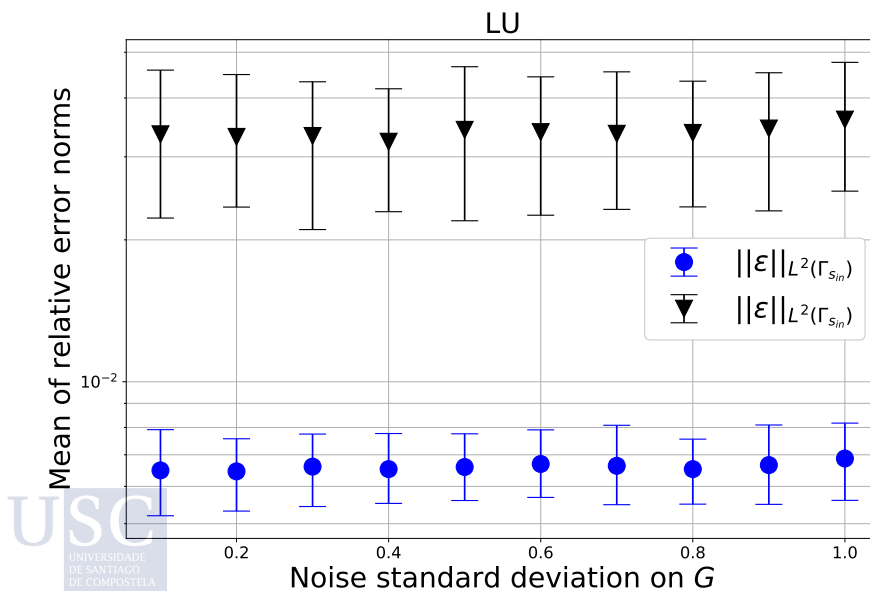


Figure 4.2.10: Effect of noise in the total heat flux measurement with 4% standard deviation noise in the thermocouples (90% quantile bars).

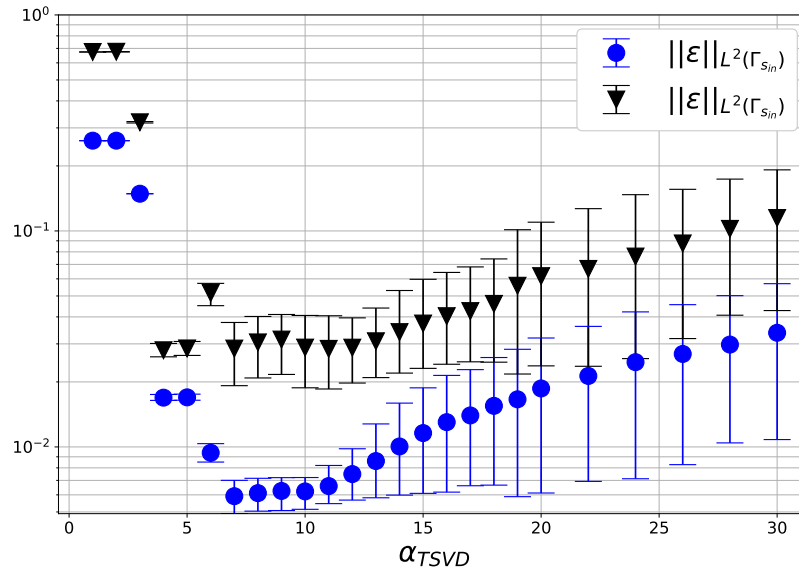


Figure 4.2.11: Effect of noise in the total heat flux measurement with 4% standard deviation noise in the thermocouples and 10% on the total heat measure (90% quantile bars).

4.2.3 Conclusions

In this benchmark case, we tested the methods presented in Chapter 3 in an industrial setting. Alifanov's regularization proved to perform very poorly. Due to the thermocouples located very close to the boundary Γ_{sin} , this regularization method overestimates the heat flux close to the measurement points, underestimating it away from the measurements. Including the total heat flux measurement in the cost functional improves the obtained results, but not to a satisfactory level.

Also in this test case, the parameterization method proved to perform very well providing excellent estimation of the heat flux. In this case, introducing the total heat measurement caused a degradation of the estimated heat flux. For this method, the TSVD regularization was used to filter the measurement noise. It allowed to obtain nice heat flux estimations also in the noisy scenario.

To conclude, Table 4.6 illustrates the CPU time required for the computations with no error in the measurements and $J_{tol} = 10^2 K^2$ in the case of only temperature measurements available. Notice that all the computations were performed in serial on a Intel[®] Core[™] i7-8550U CPU processor. Recalling that in this application the

thermocouples sample at 1 Hz , the parameterization method allows the real-time estimation of the boundary heat flux.

Table 4.6: Inverse problem CPU time comparison for the industrial benchmark case.

	Alifanov's reg.	Parameterized heat flux	
		offline	online
CPU time	221 s	121.4 s	0.15 s

5 CONCLUSIONS

We started this first part of the thesis by discussing the mold monitoring problem. Analyzing the industrial setting, we identified as the quantity of interest to determine the mold-steel heat flux. Moreover, since the objective is to monitor the mold during CC, we highlighted that this heat flux should be estimated in real-time.

The CC molds are equipped with a system that measures the cooling water temperature at the inlet and at the outlet of the cooling system, as well as the pointwise mold plates temperature thanks to some thermocouples buried into the plates. These are the data that we have available to estimate the mold heat flux.

Between all possible approaches, we used a data assimilation, optimal control one to solve this problem. In this framework, we required a mold model that has the mold-steel heat flux as parameter that we determine minimizing a distance between the measured and the simulated data.

To set up the problem, our first objective was to develop a mold model. We considered the mold plates as computational domain. Hence, the mold-steel interface is a portion of the boundary of the domain and the sought heat flux is a Neumann BC on this portion of the boundary. Notice that in this setting, the thermocouples measurements are pointwise state measurements in the interior of the domain.

We developed a mold model hierarchy by adding subsequent simplifying assumptions. In particular, the most detailed model has a domain that includes the cooling water system. It models the water flow by the Navier-Stokes equations and the heat transfer in a multiphase, advection-diffusion setting. After a number of motivated simplifying assumptions, the last model of the hierarchy is a steady-state heat conduction model in the solid portion of the mold plates with a Robin BC at the interface with the cooling water.

In formulating the boundary heat flux estimation problem, we used this latter model. Moreover, we considered two different measurement settings: having only the thermocouples pointwise temperature measurements or having them together with the total boundary heat flux. This latter measurement comes from an energy balance in the cooling water using its measured temperature increase in the cooling system. Finally, we used a deterministic least square approach in the definition of the inverse problems.

In the literature, there are several possible methods for the solution of these inverse problems. However, they generally require several iterations to estimate the

heat flux and a number of simulations of the mold model at each iteration. Due to the computational cost of these simulations, these methods are not suitable for this real-time monitoring task as they are.

To achieve our goal, we developed a novel methodology for the solution of this inverse problem. It exploits a parameterization of the heat flux. This parameterization allowed us to realize a direct method that features an offline-online decomposition. In the offline phase, we perform several computationally expensive simulations of the mold model while in the online phase we input the thermocouples measurements and compute the boundary heat flux with extremely fast computations. The advantage is that the offline phase is computed once and for all before the CC machine starts to work. Then, when the casting starts, we only need to solve the computationally cheap online phase.

Later, we compared this methodology with traditional Alifanov's regularization. We tested them in two different benchmark cases: an academic test and an industrial one. First, we validated the numerical simulation of the mold model. Notice that we used the FVM for the solution of the direct problem and all the coding was performed using the C++ library ITHACA-FV. Then, we performed several tests on the boundary heat flux reconstruction problem.

In particular, we tested the effects of the mesh refinement on the proposed methodologies, the influence of noise in the measurements, the impact of changing the thermocouples positioning and number, and the advantages of including the total heat flux in the input data. The results showed that the parameterization method outperforms Alifanov's regularization in all the tests. In fact, this method is able to estimate the boundary heat flux with much more accuracy for all the considered measurements configurations. Moreover, it showed to provide good solutions also in presence of significant noise in the measurements. Finally, it allows the real-time estimation of the boundary heat flux while Alifanov's regularization cannot be employed in real-time as it is.

As a final remark, we notice that this novel methodology can also be applied in other contexts. From the mathematical point of view, the presented industrial problem is a Neumann boundary condition estimation problem in a steady linear setting with pointwise state measurements in the interior of the domain. Then, we can employ the proposed methodology to any problem sharing these features. An example can be a boundary stress estimation problem in linear elasticity with pointwise deformation measurements.

Part II

Estimation of the Boundary Heat Flux by Using an Unsteady-State Mold Model

6 UNSTEADY-STATE DIRECT PROBLEM

We devote this second part of the thesis to the study of the mold-slab heat flux estimation problem in the unsteady-state setting. First, we discuss in the present chapter the mathematical formulation of the unsteady physical model (MU4) introduced in Section 1.3.2 and its numerical discretization. Later, we present in Chapter 7 the formulation of the boundary heat flux estimation problem and a novel methodology for its solution. To achieve real-time performances in this setting, we require the derivation of cutting-edge MOR techniques. This is the subject of Chapter 8. Finally, we design in Chapter 9 some benchmarks that we use for the validation of the direct problem solver and the testing of the proposed novel methodologies.

6.1 THREE-DIMENSIONAL UNSTEADY-STATE HEAT CONDUCTION MODEL

In the present section, we propose mathematical formulation of the unsteady-state heat conduction mold model (MU4). We derive it from model (MS4) by dropping the steady-state assumption. Thus, we refer to Section 2.6 for all the details on the domain, notation and assumptions on the data.

We shall assume all along the following assumptions on the data:

- (HU4.1) The thermal conductivity is constant and strictly positive: $k_s \in \mathbb{R}^+$.
- (HU4.2) There is no heat source inside the mold domain: $f_s = 0$.
- (HU4.3) The density and specific heat are constant and strictly positive: $\rho \in \mathbb{R}^+$, $C_p \in \mathbb{R}^+$.
- (HU4.4) The heat transfer coefficient on Γ_{sf} is constant and strictly positive: $h \in \mathbb{R}^+$.
- (HU4.5) The cooling water temperature, T_f , is known, constant in time, and belongs to $L^q(\Gamma_{sf})$.
- (HU4.6) The initial temperature, T_0 , is known and belongs to $L^2(\Omega_s)$.
- (HU4.7) The steel-mold heat flux, g , belongs to $L^r(0, t_f; L^q(\Gamma_{sin}))$.

In (HU4.5), (HU4.7) we assume that $r, q \in (2, +\infty)$ and

$$\frac{1}{r} + \frac{1}{q} < \frac{1}{2}. \quad (6.1.1)$$

Notice that it implies $r, q > 2$.

Under the assumptions (H1.1)-(H1.3), (H4.2), and (HU4.5)-(HU4.7), we propose the following unsteady-state modification of Problem 2.6

Problem 6.1. Find T_s such that

$$\rho_s C_{p_s} \frac{\partial T_s}{\partial t} - k_s \Delta T_s = 0, \quad \text{in } \Omega_s \times (0, t_f], \quad (6.1.2)$$

with BCs and Initial Condition (IC)

$$\begin{cases} -k_s \nabla T_s \cdot \mathbf{n} = g & \text{on } \Gamma_{s_{in}} \times (0, t_f], \\ -k_s \nabla T_s \cdot \mathbf{n} = 0 & \text{on } \Gamma_{s_{ex}} \times (0, t_f], \\ -k_s \nabla T_s \cdot \mathbf{n} = h(T_s - T_f) & \text{on } \Gamma_{s_f} \times (0, t_f], \\ T_s(\cdot, 0) = T_0 & \text{in } \Omega_s. \end{cases} \quad (6.1.3)$$

$$\begin{cases} -k_s \nabla T_s \cdot \mathbf{n} = 0 & \text{on } \Gamma_{s_{ex}} \times (0, t_f], \\ -k_s \nabla T_s \cdot \mathbf{n} = h(T_s - T_f) & \text{on } \Gamma_{s_f} \times (0, t_f], \\ T_s(\cdot, 0) = T_0 & \text{in } \Omega_s. \end{cases} \quad (6.1.4)$$

$$\begin{cases} -k_s \nabla T_s \cdot \mathbf{n} = h(T_s - T_f) & \text{on } \Gamma_{s_f} \times (0, t_f], \\ T_s(\cdot, 0) = T_0 & \text{in } \Omega_s. \end{cases} \quad (6.1.5)$$

$$\begin{cases} T_s(\cdot, 0) = T_0 & \text{in } \Omega_s. \end{cases} \quad (6.1.6)$$

A weak solution is now defined by testing against a smooth function and formally integrating by parts.

Definition 6.1. We say that a function $T_s \in C([0, t_f]; L^2(\Omega_s)) \cap L^2(0, t_f; H^1(\Omega_s))$ is a weak solution of Problem 6.1 on $[0, t_f]$ for some $t_f > 0$ if

$$\begin{aligned} & -\rho_s C_{p_s} \int_0^{t_f} \int_{\Omega_s} T_s(\mathbf{x}, t) \frac{\partial \psi(\mathbf{x}, t)}{\partial t} d\mathbf{x} dt + k_s \int_0^{t_f} \int_{\Omega_s} \nabla T_s(\mathbf{x}, t) \nabla \psi(\mathbf{x}, t) d\mathbf{x} dt + \\ & \int_0^{t_f} \int_{\Gamma_{s_f}} h T_s(\mathbf{x}, t) \psi(\mathbf{x}, t) d\Gamma dt = \rho_s C_{p_s} \int_{\Omega_s} T_s(\mathbf{x}) \psi(\mathbf{x}, 0) d\mathbf{x} - \\ & \int_0^{t_f} \int_{\Gamma_{s_{in}}} g(\mathbf{x}, t) \psi(\mathbf{x}, t) d\Gamma dt + \int_0^{t_f} \int_{\Gamma_{s_f}} h T_f(\mathbf{x}) \psi(\mathbf{x}, t) d\Gamma dt, \end{aligned} \quad (6.1.7)$$

for all $\psi \in H^1(0, t_f; H^1(\Omega_s))$ that satisfy $\psi(t_f) = 0$.

Thanks to Nittka [108] (its Theorem 2.11 and Corollary 2.13), we have

Theorem 6.2. Let assumptions (H1.1), (H1.2), (H4.2), (HU4.5)-(HU4.7) and (6.1.1) hold. Then, there exists a unique weak solution of Problem 6.1 on $[0, t_f]$.

Finally, we recall Theorem 3.3 in [108]

Theorem 6.3. *Let assumptions (H1.1), (H1.2), (H4.2), (HU4.5)-(HU4.7) and (6.1.1) hold. Then, the weak solution T_s of Problem 6.1 is in $C([0, t_f]; C(\bar{\Omega}_s))$. So, in particular, $T_s(\mathbf{x}, t) \rightarrow T_0(\mathbf{x})$ uniformly on $\bar{\Omega}_s$ as $t \rightarrow 0$.*

6.1.1 Numerical Discretization

Regarding the numerical solution of Problem 6.1, we use the FVM for the space discretization while considering a first-order implicit Euler scheme for the time discretization.

We start by recalling some of the notation introduced in Section 2.6.1 for the discretization of Problem 6.1. Given a tessellation \mathcal{T} of the domain, Ω_s , we denote by \mathcal{E} the set of faces of the control volumes. Given $K \in \mathcal{T}$, let $\mathcal{E}_K = \{\sigma \in \mathcal{E} | \sigma \subset \partial K\}$. For each $K \in \mathcal{T}$, $m(K)$ denotes the three-dimensional Lebesgue measure of K and \mathbf{x}_K its centroid. Analogously, for each $\sigma \in \mathcal{E}$, $m(\sigma)$ denotes the two-dimensional measure of σ and \mathbf{x}_σ its centroid. For the finite volume schemes considered here, the space discrete unknown temperature is denoted by $T_{s_K}(t)$, $K \in \mathcal{T}$. The value $T_{s_K}(t)$ is expected to be some approximation of $T_s(\mathbf{x}, t)$ in the cell K . Finally, we have

$$T_{f_\sigma} = \frac{1}{m(\sigma)} \int_\sigma T_f d\Gamma, \quad \forall \sigma \subset \Gamma_{sf}, \quad (6.1.8)$$

and, for each $t \in (0, t_f]$, we define

$$g_\sigma(t) := \frac{1}{m(\sigma)} \int_\sigma g(t) d\Gamma, \quad \forall \sigma \subset \Gamma_{sin}. \quad (6.1.9)$$

Integrating (6.1.2) over each control volume, we have

$$\int_K \rho_s C_{ps} \frac{\partial T_s(\mathbf{x}, t)}{\partial t} d\mathbf{x} + \int_K -k_s \Delta T_s(\mathbf{x}, t) d\mathbf{x} = 0, \quad \forall K \in \mathcal{T}, t \in (0, t_f]. \quad (6.1.10)$$

Applying the divergence theorem on the second term, we have

$$\int_K -k_s \Delta T_s(\mathbf{x}, t) d\mathbf{x} = \int_{\partial K} -k_s \nabla T_s(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) d\Gamma, \quad \forall K \in \mathcal{T}, t \in (0, t_f]. \quad (6.1.11)$$

Then, the mean value theorem allows us to write

$$\int_{\partial K} -k_s \nabla T_s(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) d\Gamma \approx -k_s \sum_{\sigma \in \mathcal{E}_K} (\nabla T_s)_\sigma(t) \cdot \mathbf{s}_\sigma, \quad (6.1.12)$$

where $(\nabla T_s)_\sigma(t)$ is the gradient evaluated at the face centroid at time t , and $\mathbf{s}_\sigma = \mathbf{n}_\sigma m(\sigma)$, \mathbf{n}_σ being the normal unitary vector, pointing to the outside of K , evaluated at the face centroid.

Now, we consider the control volume K_P in Figure 2.6.1 and its neighbor K_N with which it shares the σ face. Assuming an orthogonal grid and thanks to (2.6.8), we have the following second-order approximation of the gradient evaluated at the cell face

$$(\nabla T_s)_\sigma(t) \cdot \mathbf{s}_\sigma \approx |\mathbf{s}_\sigma| \frac{T_{s_N}(t) - T_{s_P}(t)}{|\mathbf{d}_{PN}|}. \quad (6.1.13)$$

By (2.6.15) and the BC (6.1.5), we can write

$$-k_s(\nabla T_s)_\sigma(t) \cdot \mathbf{s}_\sigma \approx q_{K_P,\sigma}(T_{s_P}(t) - T_{f_\sigma}), \quad \forall \sigma \in (\Gamma_{sf} \cap \partial K_P), t \in (0, t_f]. \quad (6.1.14)$$

On Γ_{sin} thanks to (6.1.3), we have

$$-k_s(\nabla T_s)_\sigma(t) \cdot \mathbf{s}_\sigma \approx m(\sigma)g_\sigma(t), \quad \forall \sigma \in (\Gamma_{sin} \cap \partial K_P), t \in (0, t_f]. \quad (6.1.15)$$

Then, for the Laplacian term in (6.1.10), we obtain

$$\begin{aligned} & \left(-k_s \sum_{\sigma \in \mathcal{E}_{K_P} \setminus \Gamma} \frac{m(\sigma)}{|\mathbf{d}_{PN}|} + \sum_{\sigma \in \mathcal{E}_{K_P} \cap \Gamma_{sf}} q_{K_P,\sigma} \right) T_{s_P}(t) = \\ & k_s \sum_{\sigma \in \mathcal{E}_{K_P} \setminus \Gamma} \frac{m(\sigma)}{|\mathbf{d}_{PN}|} T_{s_N}(t) + \sum_{\sigma \in \mathcal{E}_{K_P} \cap \Gamma_{sf}} q_{K_P,\sigma} T_{f_\sigma} - \sum_{\sigma \in \mathcal{E}_{K_P} \cap \Gamma_{sin}} m(\sigma)g_\sigma(t), \\ & \quad \forall K_P \in \mathcal{T}, t \in (0, t_f]. \end{aligned} \quad (6.1.16)$$

Discretizing in space the first term of (6.1.10), we simply have

$$\int_{K_P} \rho C_p \frac{\partial T_s(\mathbf{x}, t)}{\partial t} d\mathbf{x} \approx \rho_s C_{p_s} m(K_P) \frac{dT_{s_P}(t)}{dt}, \quad \forall K_P \in \mathcal{T}, t \in (0, t_f]. \quad (6.1.17)$$

To conclude the space discretization, if the tessellation \mathcal{T} has N_h control volumes, we introduce the time dependent vector of unknowns with dimension N_h :

$$(\mathbf{T}_s)_i(t) := T_{s_i}(t), \quad 1 \leq i \leq N_h, \quad (6.1.18)$$

the source vector

$$(\mathbf{b})_i(t) = \sum_{\sigma \in \mathcal{E}_{K_i} \cap \Gamma_{sf}} q_{K_i,\sigma} T_{f_\sigma} - \sum_{\sigma \in \mathcal{E}_{K_i} \cap \Gamma_{sin}} m(\sigma)g_\sigma(t), \quad (6.1.19)$$

the diagonal mass matrix $M \in \mathbb{M}^{N_h \times N_h}$

$$M_{ii} = m(K_i), \quad (6.1.20)$$

and the stiffness matrix $A \in \mathbb{M}^{N_h \times N_h}$, defined as

$$\begin{aligned} A_{ii} &= -k_s \sum_{\sigma \in \mathcal{E}_{K_i} \setminus \Gamma} \frac{m(\sigma)}{|\mathbf{d}_{PN}|} + \sum_{\sigma \in \mathcal{E}_{K_i} \cap \Gamma_{sf}} q_{K_i, \sigma}, \\ A_{ij} &= -k_s \frac{m(\sigma)}{|\mathbf{d}_{ij}|}, && \text{if there exists } \sigma \in \partial K_i \cap \partial K_j \\ A_{ij} &= 0, && \text{otherwise.} \end{aligned} \quad (6.1.21)$$

Notice that we consider all vectors to be column vectors.

So, after the space discretization, (6.1.10) is rewritten as

$$\rho_s C_{p_s} M \frac{d\mathbf{T}_s(t)}{dt} + A \mathbf{T}_s(t) = \mathbf{b}(t), \quad t \in (0, t_f]. \quad (6.1.22)$$

To discretize (6.1.22) in time, we divide the time interval of interest into N_T regular steps

$$t^0 = 0, \quad t^{n+1} = t^n + \Delta t, \quad n = 0, \dots, N_T - 1, \quad \Delta t = \frac{t_f}{N_T}. \quad (6.1.23)$$

From now on, we denote by f^j an approximation of a given function $f(t)$ at time t^j . For the time discretization, we consider the first-order, implicit Euler scheme. Integrating (6.1.22) from t^n to $t^{n+1} = t^n + \Delta t$ yields

$$\rho_s C_{p_s} M [\mathbf{T}_s(t^{n+1}) - \mathbf{T}_s(t^n)] = \int_{t^n}^{t^{n+1}} \mathbf{b}(t) - A \mathbf{T}_s(t) dt. \quad (6.1.24)$$

Now, approximating the integral on the right by the right-hand rectangle method, we have

$$\rho_s C_{p_s} M [\mathbf{T}_s(t^{n+1}) - \mathbf{T}_s(t^n)] \approx \Delta t \mathbf{b}(t^{n+1}) - \Delta t A \mathbf{T}_s(t^{n+1}). \quad (6.1.25)$$

Finally, since \mathbf{T}_s^n is an approximation of $\mathbf{T}_s(t^{n+1})$ and rearranging the terms, we obtain the implicit Euler scheme

$$(\rho_s C_{p_s} M + \Delta t A) \mathbf{T}_s^{n+1} = \rho_s C_{p_s} M \mathbf{T}_s^n + \Delta t \mathbf{b}^{n+1}, \quad n = 0, \dots, N_T - 1. \quad (6.1.26)$$

Notice that, thanks to hypotheses (HU4.1)-(HU4.7), matrices A and M are time independent.

This is a general overview on the finite volume discretization of Problem 6.1 assuming an orthogonal structured grid. The actual value of each element of A and \mathbf{b} depends on the mesh used and the possible correction terms due to non orthogonalities in the mesh. Since our problem is a classic diffusion problem, we refer for further details on this finite volume discretization to the Eymard's monograph [52].

7 UNSTEADY-STATE INVERSE PROBLEM

We devote the present chapter to the formulation and solution of the mold-steel heat flux estimation problem in the unsteady-state case. We start by discussing the mathematical formulation of the problem in Section 7.1. Here we state two different inverse problems. Thus, we devote Section 7.2 and 7.3 to the development of some novel methodologies for their solution.

7.1 INVERSE THREE-DIMENSIONAL UNSTEADY-STATE HEAT CONDUCTION PROBLEM

Before proceeding with the mathematical formulation of the inverse problem, we do some technical considerations that will guide us in the process. First, the thermocouples measure the temperature at the sampling frequency f_{samp} . This sampling frequency is typically of 1 *Hz* and we will assume this value all along this investigation (notice that different values of f_{samp} are compatible with the following discussion). Second, every sampling period $T_{samp} = 1/f_{samp} = 1$ s, the thermocouples provide a new set of measurements, so we have a regular sequence of measurements in time.

That said, we consider the problem of estimating the heat flux, g , on Γ_{sin} , in between the last acquired measurement instant and the previous one. In this way, we follow the sequentiality of the measured data in our solution procedure according to the real-time purpose of this research.

We introduce the following notation. Let $\Upsilon := \{\tau^0, \tau^1, \dots, \tau^{P_t}\}$ be a collection of points in $[0, t_f]$ such that $\tau^k = t^{kN_t}$ (see Figure 7.1.1). According to the introduced sequential approach, we consider the following restriction of Problem 6.1 to $(\tau^{k-1}, \tau^k]$, $1 \leq k \leq P_t$, as direct problem

Problem 7.1. *Let $1 \leq k \leq P_t$ and $g^k(\mathbf{x}, t)$ be a given heat flux on $\Gamma_{sin} \times (\tau^{k-1}, \tau^k]$. Find T_s^k such that*

$$\rho_s C_{ps} \frac{\partial T_s^k}{\partial t} - k_s \Delta T_s^k = 0, \quad \text{in } \Omega_s \times (\tau^{k-1}, \tau^k], \quad (7.1.1)$$

with BCs and IC

$$\begin{cases} -k_s \nabla T_s^k \cdot \mathbf{n} = g^k & \text{on } \Gamma_{sin} \times (\tau^{k-1}, \tau^k], & (7.1.2) \\ -k_s \nabla T_s^k \cdot \mathbf{n} = 0 & \text{on } \Gamma_{sex} \times (\tau^{k-1}, \tau^k], & (7.1.3) \\ -k_s \nabla T_s^k \cdot \mathbf{n} = h(T_s^k - T_f) & \text{on } \Gamma_{sf} \times (\tau^{k-1}, \tau^k], & (7.1.4) \\ T_s^k(\cdot, \tau^{k-1}) = T_s^{k-1}(\cdot, \tau^{k-1}) & \text{in } \Omega_s, & (7.1.5) \end{cases}$$

where $T_s^0(\cdot, \tau^0) = T_0$, being T_0 the initial temperature.

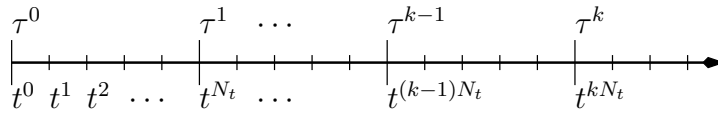


Figure 7.1.1: Time line for the inverse problem.

So basically, we are dividing the time domain into chunks going from one measurement time to the next one in a way that facilitates the definition of the inverse problems below. Before formulating it, we introduce some further notation. We define the application $(\mathbf{x}_i, \tau^k) \in \Psi \times \Upsilon \rightarrow \hat{T}(\mathbf{x}_i, \tau^k) \in \mathbb{R}^+$, $1 \leq i \leq P$, $1 \leq k \leq P_t$, $\hat{T}(\mathbf{x}_i, \tau^k)$ being the experimentally measured temperature at $(\mathbf{x}_i, \tau^k) \in \Psi \times \Upsilon$. Moreover, to simplify the notation, and if there is no room for error, we denote

$$\hat{T}^k(\mathbf{x}_i) := \hat{T}(\mathbf{x}_i, \tau^k), \quad 1 \leq i \leq P, \quad 1 \leq k \leq P_t, \quad (7.1.6)$$

and we let $T_s^k[g]$ represent the solution of Problem 7.1 corresponding to heat flux g on $\Gamma_{sin} \times (\tau^{k-1}, \tau^k]$.

At each measurement interval k , $1 \leq k \leq P_t$, we propose an iterative procedure, assuming that, for $k \geq 1$, g^l and $T_s^l[g^l]$, $0 \leq l \leq k - 1$, have been computed. Using a least square, deterministic approach, we state two different inverse problems for Problem 7.1. In the first one, we consider as functional to be minimized a distance between the measured and computed temperatures at the thermocouples. Then, we state it as

Problem 7.2. (Inverse) Being g^l and $T_s^l[g^l]$, $1 \leq l \leq k - 1$, known, and given the temperature measurements $\hat{T}^k(\mathbf{x}_i)$, $1 \leq i \leq P$, find $g^k \in L^r(\tau^{k-1}, \tau^k; L^q(\Gamma_{sin}))$ which minimizes the functional

$$S_1^k[g^k] = \frac{1}{2} \sum_{i=1}^P [T_s^k[g^k](\mathbf{x}_i, \tau^k) - \hat{T}^k(\mathbf{x}_i)]^2, \quad (7.1.7)$$

with r, q verifying (6.1.1). Here, we denote $T_s^0[g^0] = T_0$.

The second inverse problem that we consider includes in the cost functional the L^2 -norm of the heat flux. Thus, we write it as

Problem 7.3. (Inverse) *Being g^l and $T_s^l[g^l]$, $1 \leq l \leq k - 1$, known, and given the temperature measurements $\hat{T}^k(\mathbf{x}_i)$, $1 \leq i \leq P$, find $g^k \in L^r(\tau^{k-1}, \tau^k; L^q(\Gamma_{s_{in}}))$ which minimizes the functional*

$$S_2^k[g^k] = \frac{1}{2} \sum_{i=1}^P [T_s^k[g^k](\mathbf{x}_i, \tau^k) - \hat{T}^k(\mathbf{x}_i)]^2 + p_g \langle g^k(\tau^k), g^k(\tau^k) \rangle_{L^2(\Gamma_{s_{in}})}, \quad (7.1.8)$$

where $p_g \left[\frac{K^2}{W^2} \right]$ is a weight applied to the heat flux norm.

7.2 INVERSE SOLVER FOR S_1^k

In this section, we discuss a novel methodology for solving Problem 7.2. In particular, we mimic the methodology developed for the steady case in Section 3.1.2, expanding it to the unsteady case.

As in the steady case, we exploit a suitable parameterization of the heat flux, g^k . To properly parameterize it, we start by considering that we want to parameterize an unknown function g^k in $L^r(\tau^{k-1}, \tau^k; L^q(\Gamma_{s_{in}}))$, $1 \leq k \leq P_t$. We decide to split the parameterization of the heat flux between time and space. For the parameterization in space, we use the same RBF technique described in Section 3.1.2.

Thus, the parameterization of the boundary heat flux reads (see Prando's appendix [120])

$$g^k(\mathbf{x}, t) \approx \mathbf{g}^k(\mathbf{x}, t) = \sum_{i=1}^P g_i^k(t) \phi_i(\mathbf{x}), \quad \text{for } t \in (\tau^{k-1}, \tau^k], \quad (7.2.1)$$

where the $\phi_i(\mathbf{x})$ are P known basis functions as defined in (3.1.24), and the $g_i^k(t)$ are the respective time dependent unknown weights. Note that the methodology is very well adapted to the application in use, because the number of base elements that we consider matches the number of thermocouples. For a detailed discussion on the space parameterization of the heat flux, we redirect the reader to our previous discussion as the beginning of Section 3.1.2.

In this work, we investigate two different approaches for the parameterization of g^k . In the first one, we consider g_i^k independent of time

$$g_i^k(t) = w_i^k, \quad \text{for } t \in (\tau^{k-1}, \tau^k], 1 \leq i \leq P, \quad (7.2.2)$$

being w_i^k real numbers. In this way, the heat flux is assumed to be piecewise constant, i.e. constant between consecutive measurement instants.

The second approach considers the heat flux to be continuous piecewise linear in $(0, t_f]$, being a polynomial of degree 1 between the sampling times. Then, we assume the weights $g_i^k(t)$ to be linear in time in the interval $(\tau^{k-1}, \tau^k]$. Moreover, in this second case, the following continuity is assumed

$$g_i^k(t)|_{t \downarrow \tau^k} = g_i^{k-1}(t)|_{t \uparrow \tau^k}. \quad (7.2.3)$$

In turn, we characterize $g_i^k(t)$ as

$$g_i^k(t) = w_i^{k-1} + (t - \tau^{k-1}) \frac{w_i^k - w_i^{k-1}}{\tau^k - \tau^{k-1}}, \quad \text{for } t \in (\tau^{k-1}, \tau^k]. \quad (7.2.4)$$

Notice that by doing these parameterizations, we change the problem from estimating a function in an infinite dimensional space at each time interval $(t^{(k-1)N_t}, t^{kN_t}] = (\tau^{k-1}, \tau^k]$, to estimating the vector $\mathbf{w}^k = (w_1^k, w_2^k, \dots, w_P^k)^T$ in \mathbb{R}^P , for each $1 \leq k \leq P_t$.

Now, at each time interval $(\tau^{k-1}, \tau^k]$, the objective of the inverse problem is to determine \mathbf{w}^k which identifies \mathbf{g}^k once the elements of the basis ϕ_i , $i = 1, 2, \dots, P$ are fixed. We state the inverse problem as

Problem 7.4. (Inverse) Given the temperature measurements $\hat{T}(\Psi, \tau^k)$, find $\hat{\mathbf{w}}^k \in \mathbb{R}^P$, $1 \leq k \leq P_t$, which minimizes the functional

$$S_1^k[\mathbf{w}^k] = \frac{1}{2} \sum_{i=1}^P [T_s^k[\mathbf{w}^k](\mathbf{x}_i, \tau^k) - \hat{T}^k(\mathbf{x}_i)]^2, \quad (7.2.5)$$

where, if there is not room for confusion, $T_s^k[\mathbf{w}^k]$ denotes the temperature $T_s^k[\mathbf{g}^k]$, with \mathbf{g}^k defined as in (7.2.1) and $g_i^k(t)$ given by (7.2.2) or (7.2.4).

For a later use, we define the general vector \mathbf{a}^k of \mathbb{R}^P as the vector of the values of a general field $a(\mathbf{x}, t)$ at the measurement points and at the measurement time τ^k , such as



$$(\mathbf{a}^k)_i = a(\mathbf{x}_i, \tau^k). \quad (7.2.6)$$

Moreover, given \mathbf{w}^k , we define the residual vector $\mathbf{R}^k[\mathbf{w}^k] \in \mathbb{R}^P$ as

$$(\mathbf{R}^k[\mathbf{w}^k])_i := (\mathbf{T}_s^k[\mathbf{w}^k])_i - (\hat{\mathbf{T}}^k)_i, \quad i = 1, 2, \dots, P. \quad (7.2.7)$$

Thanks to (7.2.7), we rewrite the cost functional (7.2.5) as

$$S_1^k[\mathbf{w}^k] = \frac{1}{2} \mathbf{R}^k[\mathbf{w}^k]^T \mathbf{R}^k[\mathbf{w}^k]. \quad (7.2.8)$$

To minimize it, we write the critical point equation

$$\frac{\partial S_1^k[\hat{\mathbf{w}}^k]}{\partial w_j^k} = \sum_{i=1}^P (R^k[\hat{\mathbf{w}}^k])_i \frac{\partial (\mathbf{T}_s^k[\hat{\mathbf{w}}^k])_i}{\partial w_j^k} = 0, \text{ for } j = 1, 2, \dots, P. \quad (7.2.9)$$

Thus, for each k , $1 \leq k \leq P_t$, the solution of this equation will provide the weights vector $\hat{\mathbf{w}}^k$ corresponding to a critical point of S_1^k .

To explicitly obtain from (7.2.9) an equation for the weights that minimizes our functional S_1^k , we exploit the linearity of Problem 7.1. To derive it, we consider first the piecewise constant (7.2.2), and then the piecewise linear (7.2.4) cases.

7.2.1 Piecewise Constant Approximation of the Heat Flux

Suppose to have the solutions to the following auxiliary problems

Problem 7.5. For each i , $1 \leq i \leq P$, find T_{ϕ_i} such that

$$\rho_s C_{ps} \frac{\partial T_{\phi_i}}{\partial t} - k_s \Delta T_{\phi_i} = 0, \quad \text{in } \Omega_s \times (\tau^0, \tau^1], \quad (7.2.10)$$

with BCs and IC

$$\begin{cases} -k_s \nabla T_{\phi_i} \cdot \mathbf{n} = \phi_i & \text{on } \Gamma_{sin} \times (\tau^0, \tau^1], & (7.2.11) \\ -k_s \nabla T_{\phi_i} \cdot \mathbf{n} = 0 & \text{on } \Gamma_{sex} \times (\tau^0, \tau^1], & (7.2.12) \\ -k_s \nabla T_{\phi_i} \cdot \mathbf{n} = h T_{\phi_i} & \text{on } \Gamma_{sf} \times (\tau^0, \tau^1], & (7.2.13) \\ T_{\phi_i}(\cdot, \tau^0) = 0 & \text{in } \Omega_s. & (7.2.14) \end{cases}$$

Problem 7.6. For each k , $1 \leq k \leq P_t$, find T_{IC}^k such that

$$\rho_s C_{ps} \frac{\partial T_{IC}^k}{\partial t} - k_s \Delta T_{IC}^k = 0, \quad \text{in } \Omega_s \times (\tau^{k-1}, \tau^k], \quad (7.2.15)$$

with BCs and IC

$$\begin{cases} -k_s \nabla T_{IC}^k \cdot \mathbf{n} = 0 & \text{on } (\Gamma_{sin} \cup \Gamma_{sex}) \times (\tau^{k-1}, \tau^k] & (7.2.16) \\ -k_s \nabla T_{IC}^k \cdot \mathbf{n} = h (T_{IC}^k - T_f) & \text{on } \Gamma_{sf} \times (\tau^{k-1}, \tau^k], & (7.2.17) \\ T_{IC}^k(\cdot, \tau^{k-1}) = T_s^{k-1}(\cdot, \tau^{k-1}) & \text{in } \Omega_s, & (7.2.18) \end{cases}$$

with

$$T_{IC}^1(\cdot, \tau^0) = T_0. \quad (7.2.19)$$

Notice that Problem 7.5 does not depend on the measurement instants index k . Then, we define it only in the first interval $(\tau^0, \tau^1]$ and, if needed, we translate it such as

$$T_{\phi_i}^k(\mathbf{x}, t) = T_{\phi_i}(\mathbf{x}, t - \tau^{k-1}), \text{ for } t \in (\tau^{k-1}, \tau^k]. \quad (7.2.20)$$

We can now state

Theorem 7.1. *Given T_{IC}^k and \mathbf{w}^k , $1 \leq k \leq P_t$ (and so \mathbf{g}^k defined as (7.2.1)) and T_{ϕ_i} , $1 \leq i \leq P$, the function defined as*

$$T_s^k[\mathbf{g}^k] = \sum_{i=1}^P g_i^k T_{\phi_i}^k + T_{IC}^k, \quad (7.2.21)$$

is the solution to Problem 7.1 associated with the heat flux $\mathbf{g}(x, t)$ which in each measurement subinterval $(\tau^{k-1}, \tau^k]$ coincides with $\mathbf{g}^k(x, t)$ given by (7.2.1) with $g_i^k(t)$ as in (7.2.2).

Proof. Substituting (7.2.21) in (7.1.1) and considering (7.2.10) and (7.2.15), we have

$$\begin{aligned} \rho_s C_{p_s} \frac{\partial \left(\sum_{i=1}^P g_i^k(t) T_{\phi_i}^k + T_{IC}^k \right)}{\partial t} - k_s \Delta \left(\sum_{i=1}^P g_i^k(t) T_{\phi_i}^k + T_{IC}^k \right) = \\ \sum_{i=1}^P g_i^k(t) \left(\rho_s C_{p_s} \frac{\partial T_{\phi_i}^k}{\partial t} - k_s \Delta T_{\phi_i}^k \right) + \rho_s C_{p_s} \frac{\partial T_{IC}^k}{\partial t} - k_s \Delta T_{IC}^k = 0 \end{aligned} \quad (7.2.22)$$

in $\Omega_s \times (\tau^{k-1}, \tau^k]$.

Similarly, for the BCs we have

$$-k_s \nabla \left(\sum_{i=1}^P g_i^k T_{\phi_i}^k + T_{IC}^k \right) \cdot \mathbf{n} = \sum_{i=1}^P g_i^k \left(-k_s \nabla T_{\phi_i}^k \cdot \mathbf{n} \right) = \quad (7.2.23)$$



$$\sum_{i=1}^P g_i^k \phi_i = \mathbf{g}^k, \text{ on } \Gamma_{s_{in}} \times (\tau^{k-1}, \tau^k],$$

$$-k_s \nabla \left(\sum_{i=1}^P g_i^k T_{\phi_i}^k + T_{IC}^k \right) \cdot \mathbf{n} = 0, \text{ on } \Gamma_{s_{ex}} \times (\tau^{k-1}, \tau^k], \quad (7.2.24)$$

and

$$\begin{aligned}
 -k_s \nabla \left(\sum_{i=1}^P g_i^k T_{\phi_i}^k + T_{IC}^k \right) \cdot \mathbf{n} &= h \left[\sum_{i=1}^P g_i^k T_{\phi_i}^k + T_{IC}^k - T_f \right] = \\
 &h \left(T_s^k[\mathbf{g}^k] - T_f \right), \text{ on } \Gamma_{sf} \times (\tau^{k-1}, \tau^k].
 \end{aligned} \tag{7.2.25}$$

With respect to the IC, at each interval we must proceed by induction. For $k = 1$,

$$T_s^1[g^1](\cdot, \tau^0) = T^0(\cdot). \tag{7.2.26}$$

For $k > 1$,

$$\begin{aligned}
 T_s^k[g^k](\cdot, \tau^{k-1}) &= \sum_{i=1}^P g_i^k(\tau^{k-1}) T_{\phi_i}^k(\cdot, \tau^{k-1}) + \\
 T_{IC}^k(\cdot, \tau^{k-1}) &= T_s^{k-1}(\cdot, \tau^{k-1}), \text{ in } \Omega_s.
 \end{aligned} \tag{7.2.27}$$

This ends the proof. \square

Thanks to (7.2.21), (7.2.9) can be written as

$$\begin{aligned}
 \sum_{i=1}^P (R^k[\hat{\mathbf{w}}^k])_i \frac{\partial \left(\sum_{l=1}^P g_l^k \mathbf{T}_{\phi_l} + \mathbf{T}_{IC}^k \right)_i}{\partial w_j^k} &= \\
 \sum_{i=1}^P (R^k[\hat{\mathbf{w}}^k])_i \frac{\partial \left(\sum_{l=1}^P w_l^k \mathbf{T}_{\phi_l} + \mathbf{T}_{IC}^k \right)_i}{\partial w_j^k} &= \\
 \mathbf{R}^k[\hat{\mathbf{w}}^k]^T (\mathbf{T}_{\phi_j}) &= 0, \quad \text{for } j = 1, 2, \dots, P,
 \end{aligned} \tag{7.2.28}$$

where $(\mathbf{T}_{\phi_l})_j$ is the vector containing the values of the field $T_{\phi_l}(\mathbf{x}_j, \tau_1)$ at the measurement points. We recall that \mathbf{T}_{ϕ_l} is independent of k .

Let us define the matrix Θ in $\mathbb{M}^{P \times P}$ such that

$$\Theta_{i,j} = T_{\phi_j}(\mathbf{x}_i, \tau^1). \tag{7.2.29}$$

Equation (7.2.28) can now be written as

$$\Theta^T \mathbf{R}^k[\hat{\mathbf{w}}^k] = \mathbf{0}. \tag{7.2.30}$$

Using (7.2.21), the vector associated to the solution of the direct problem at the measurement points, $\mathbf{T}_s^k[\mathbf{w}^k] \in \mathbb{R}^P$, for each k , can be written as

$$\mathbf{T}_s^k[\mathbf{w}^k] = \sum_{j=1}^P w_j^k \mathbf{T}_{\phi_j} + \mathbf{T}_{IC}^k = \Theta \mathbf{w}^k + \mathbf{T}_{IC}^k. \tag{7.2.31}$$

Recalling the definition of \mathbf{R}^k and (7.2.31), we have

$$\Theta^T \mathbf{R}^k[\hat{\mathbf{w}}^k] = \Theta^T (\Theta \hat{\mathbf{w}}^k + \mathbf{T}_{IC}^k - \hat{\mathbf{T}}^k) = \mathbf{0}. \quad (7.2.32)$$

Therefore, for each k , $1 \leq k \leq P_t$, a solution of the inverse problem, $\hat{\mathbf{w}}^k$, is obtained by solving the linear system

$$\Theta^T \Theta \hat{\mathbf{w}}^k = \Theta^T (\hat{\mathbf{T}}^k - \mathbf{T}_{IC}^k). \quad (7.2.33)$$

It is important to notice that the system matrix, $\Theta^T \Theta$, is k -independent.

Equation (7.2.33) is generally called the normal equation. By solving this linear system, we obtain the weights, $\hat{\mathbf{w}}^k$, that correspond to a critical point of the functional (7.2.5). As mentioned, Θ is constant. So it can be computed once and for all in an offline phase.

The proposed methodology for the solution of the inverse Problem 7.2 is summarized in Algorithm 3. It is important to notice that, for each time interval $(\tau^{k-1}, \tau^k]$, T_{ϕ_i} and the related vector do not change but T_{IC}^k does because its initial condition depends on the temperature field at time τ^{k-1} . So, it has to be computed during the online phase.

Notice that, in this setting, (7.2.33) is an affine map from the observations, $\hat{\mathbf{T}}^k$, to the heat flux weights, $\hat{\mathbf{w}}^k$. Consequently, we have that the existence and uniqueness of the solution of the inverse problem depends on the invertibility of the matrix $\Theta^T \Theta$. We can easily see that the matrix is symmetric and positive semi-definite. In general, however, we cannot ensure that it is invertible. In fact, the invertibility depends on the choice of the basis functions, the computational domain and the BCs.

Before moving to the piecewise linear case, we highlight that we divided Algorithm 3 into an offline (expensive) phase and an online (cheap) phase. In the offline phase, we compute T_{ϕ_i} for $i = 1, 2, \dots, P$ by solving Problem 7.5 and assemble the related matrix Θ . Then, in the online phase, we input the measurements $\hat{\mathbf{T}}$, solve Problem 7.6 and the linear system (7.2.33). Moreover, to further reduce the computational cost of the online phase, we can perform during the offline phase a factorization (e.g. LU, QR, etc.) of the matrix of the linear system (7.2.33) so that, in the online phase, the solution of the linear system is much faster.

For the choice made when selecting the basis functions, this linear system has the dimensions of the number of thermocouples (quite small, in general). However, solution of Problem 7.6 involves the solution of a Full Order Model (FOM). As a consequence, this method is not suitable for real-time as it is. To achieve real-time performances, we apply MOR techniques. This is the subject of Chapter 8.

As a final remark, we notice that for the application of this method linearity of the direct problem is essential. In fact, it is a necessary condition for Theorem 7.1.

Algorithm 3 Inverse solver for the solution of Problem 7.2 with piecewise constant parameterization in time of the heat flux, g .

OFFLINE

Input RBF shape parameter, η ; thermocouples measurement points and times, Ψ, Υ

- 1: Setup RBF parameterization by (3.1.24)
- 2: Compute T_{ϕ_i} for $i = 1, 2, \dots, P$ by solving Problem 7.5
- 3: Assemble matrix Θ by (7.2.29)

ONLINE

Input Initial condition, T_0

- 1: Set $k = 1$
- 2: **while** $k \leq P_t$ **do**
- 3: Read the thermocouples measurements, $\hat{\mathbf{T}}^k$
- 4: Compute T_{IC}^k by solving Problem 7.6
- 5: Assemble \mathbf{T}_{IC}^k
- 6: Compute $\hat{\mathbf{w}}^k$ by solving (7.2.33)
- 7: Compute $g_i^k(t)$ by (7.2.2)
- 8: Compute the heat flux $\mathbf{g}(\mathbf{x}, t)$ for $t \in (\tau^{k-1}, \tau^k]$ by (7.2.1) and (7.2.2)
- 9: Use (7.2.21) to compute $T_s^k[\mathbf{g}^k]$
- 10: $k = k + 1$
- 11: **end while**

7.2.2 Piecewise Linear Heat Flux

Suppose to have the solution to

Problem 7.7. For each i , $1 \leq i \leq P$, find T_{d_i} such that

$$\rho_s C_{p_s} \frac{\partial T_{d_i}}{\partial t} - k_s \Delta T_{d_i} = -\rho_s C_{p_s} T_{\phi_i}, \quad \text{in } \Omega_s \times (\tau^0, \tau^1], \quad (7.2.34)$$

with BCs and IC



$$\begin{cases} -k_s \nabla T_{d_i} \cdot \mathbf{n} = 0 & \text{on } (\Gamma_{sin} \cup \Gamma_{sex}) \times (\tau^0, \tau^1], & (7.2.35) \\ -k_s \nabla T_{d_i} \cdot \mathbf{n} = h T_{d_i} & \text{on } \Gamma_{sf} \times (\tau^0, \tau^1], & (7.2.36) \\ T_{d_i}(\cdot, \tau^0) = 0 & \text{in } \Omega_s. & (7.2.37) \end{cases}$$

From the solution of this problem, we define

$$T_{d_i}^k(\mathbf{x}, t) = T_{d_i}(\mathbf{x}, t - \tau^{k-1}), \text{ for } t \in (\tau^{k-1}, \tau^k]. \quad (7.2.38)$$

Then, we state

Theorem 7.2. *Given T_{IC}^k and $\mathbf{w}^k, 1 \leq k \leq P_t$ (and so \mathbf{g}^k defined as (7.2.1)), T_{d_i} and $T_{\phi_i}, 1 \leq i \leq P$, then the function defined as*

$$T_s^k[\mathbf{g}^k] = \sum_{i=1}^P \left(g_i^k T_{\phi_i}^k + g_i^{k'} T_{d_i}^k \right) + T_{IC}^k, \quad (7.2.39)$$

is the solution to Problem 7.1 associated with the heat flux $\mathbf{g}(x, t)$ which in each measurement subinterval $(\tau^{k-1}, \tau^k]$ coincides with $\mathbf{g}^k(x, t)$ given by (7.2.1), with $g_i^k(t)$ as in (7.2.4).

Proof. Substituting (7.2.39) in (7.1.1) and considering (7.2.10) and (7.2.15), we have

$$\begin{aligned} \rho_s C_{p_s} \frac{\partial \left[\sum_{i=1}^P \left(g_i^k(t) T_{\phi_i}^k + g_i^{k'} T_{d_i}^k \right) + T_{IC}^k \right]}{\partial t} - k_s \Delta \left[\sum_{i=1}^P \left(g_i^k(t) T_{\phi_i}^k + g_i^{k'} T_{d_i}^k \right) + \right. \\ \left. T_{IC}^k \right] = \rho_s C_{p_s} \left[\sum_{i=1}^P g_i^{k'} T_{\phi_i}^k + \sum_{i=1}^P g_i^k(t) \left(\frac{\partial T_{\phi_i}^k}{\partial t} - k_s \Delta T_{\phi_i}^k \right) \right] + \\ \sum_{i=1}^P g_i^{k'} \left(\rho_s C_{p_s} \frac{\partial T_{d_i}^k}{\partial t} - k_s \Delta T_{d_i}^k \right) + \rho_s C_{p_s} \frac{\partial T_{IC}^k}{\partial t} - k_s \Delta T_{IC}^k = \\ \rho_s C_{p_s} \sum_{i=1}^P g_i^{k'} T_{\phi_i}^k + \sum_{i=1}^P g_i^{k'} \left(\rho_s C_{p_s} \frac{\partial T_{d_i}^k}{\partial t} - k_s \Delta T_{d_i}^k \right) = 0, \\ \text{in } \Omega_s \times (\tau^{k-1}, \tau^k]. \end{aligned} \quad (7.2.40)$$

Similarly, for the BCs we have

$$\begin{aligned} -k_s \nabla \cdot \left(\sum_{i=1}^P \left(g_i^k T_{\phi_i}^k + g_i^{k'} T_{d_i}^k \right) + T_{IC}^k \right) \cdot \mathbf{n} = \\ \sum_{i=1}^P g_i^k \left(-k_s \nabla T_{\phi_i}^k \cdot \mathbf{n} \right) = \sum_{i=1}^P g_i^k \phi_i = \mathbf{g}^k, \text{ on } \Gamma_{s_{in}} \times (\tau^{k-1}, \tau^k], \end{aligned} \quad (7.2.41)$$

$$-k_s \nabla \left(\sum_{i=1}^P \left(g_i^k T_{\phi_i}^k + g_i^{k'} T_{d_i}^k \right) + T_{IC}^k \right) \cdot \mathbf{n} = 0 \text{ on } \Gamma_{sex} \times (\tau^{k-1}, \tau^k], \quad (7.2.42)$$

and

$$\begin{aligned} & -k_s \nabla \left(\sum_{i=1}^P \left(g_i^k T_{\phi_i}^k + g_i^{k'} T_{d_i}^k \right) + T_{IC}^k \right) \cdot \mathbf{n} = \\ & h \left[\sum_{i=1}^P \left(g_i^k T_{\phi_i}^k + g_i^{k'} T_{d_i}^k \right) + T_{IC}^k - T_f \right] = h \left(T^k[\mathbf{g}^k] - T_f \right), \quad (7.2.43) \\ & \text{on } \Gamma_{sf} \times (\tau^{k-1}, \tau^k]. \end{aligned}$$

With respect to the IC, at each interval we must proceed by induction. For $k = 1$,

$$T_s^1[\mathbf{g}^1](\cdot, \tau^0) = T^0(\cdot). \quad (7.2.44)$$

For $k > 1$,

$$\begin{aligned} T_s^k[\mathbf{g}^k](\cdot, \tau^{k-1}) &= \sum_{i=1}^P \left(g_i^k(\tau^{k-1}) T_{\phi_i}^k(\cdot, \tau^{k-1}) + g_i^{k'} T_{d_i}^k(\cdot, \tau^{k-1}) \right) + \\ T_{IC}^k(\cdot, \tau^{k-1}) &= T_s^{k-1}(\cdot, \tau^{k-1}), \text{ in } \Omega_s. \end{aligned} \quad (7.2.45)$$

□

Thanks to Theorem 7.2, we have

$$\begin{aligned} \frac{\partial \mathbf{T}_s^k[\mathbf{w}^k]}{\partial w_j^k} &= \frac{\partial \left[\sum_{l=1}^P \left(g_l^k(\tau^k) \mathbf{T}_{\phi_l}^k(\tau^k) + g_l^{k'} \mathbf{T}_{d_l}^k(\tau^k) \right) + \mathbf{T}_{IC}^k(\tau^k) \right]}{\partial w_j^k} = \\ & \frac{\partial \left\{ \sum_{l=1}^P \left[w_l^k \mathbf{T}_{\phi_l}^k(\tau^k) + (w_l^k - w_l^{k-1}) f_{samp} \mathbf{T}_{d_l}^k(\tau^k) \right] + \mathbf{T}_{IC}^k(\tau^k) \right\}}{\partial w_j^k} = \\ & \mathbf{T}_{\phi_j}^k(\tau^k) + f_{samp} \mathbf{T}_{d_j}^k(\tau^k) = \mathbf{T}_{\phi_j}(\tau^1) + f_{samp} \mathbf{T}_{d_j}(\tau^1). \end{aligned} \quad (7.2.46)$$

Thanks to (7.2.46), (7.2.9) in the linear case is rewritten as

$$\mathbf{R}^k[\hat{\mathbf{w}}^k]^T (\mathbf{T}_{\phi_j} + f_{samp} \mathbf{T}_{d_j}) = 0, \text{ for all } j = 1, 2, \dots, P. \quad (7.2.47)$$

Let us define the matrices $\tilde{\Theta}, \Theta_d$ in $\mathbb{M}^{P \times P}$ such that

$$(\Theta_d)_{i,j} := f_{samp} T_{d_j}(\mathbf{x}_i, \tau^1), \quad (\tilde{\Theta})_{i,j} := (\Theta)_{i,j} + (\Theta_d)_{i,j}, \quad (7.2.48)$$

independent of the index k .

Using (7.2.48) and (7.2.39), we have

$$\begin{aligned} \mathbf{T}_s^k[\hat{\mathbf{w}}^k](\tau^k) &= \sum_{l=1}^P \left(g_l^k(\tau^k) \mathbf{T}_{\phi_l} + g_l^{k'} \mathbf{T}_{d_l} \right) + \mathbf{T}_{IC}^k(\tau^k) = \\ &= \sum_{l=1}^P \left[w_l^k \mathbf{T}_{\phi_l} + (w_l^k - w_l^{k-1}) f_{samp} \mathbf{T}_{d_l} \right] + \mathbf{T}_{IC}^k(\tau^k) = \\ &= \tilde{\Theta} \hat{\mathbf{w}}^k - \Theta_d \hat{\mathbf{w}}^{k-1} + \mathbf{T}_{IC}^k(\tau^k). \end{aligned} \quad (7.2.49)$$

Recalling the definition of \mathbf{R}^k , taking into account (7.2.48) and (7.2.49), system (7.2.47) can be written as

$$(\tilde{\Theta})^T \mathbf{R}^k[\hat{\mathbf{w}}^k] = (\tilde{\Theta})^T \left(\tilde{\Theta} \hat{\mathbf{w}}^k - \Theta_d \hat{\mathbf{w}}^{k-1} + \mathbf{T}_{IC}^k(\tau^k) - \hat{\mathbf{T}}^k \right) = \mathbf{0}. \quad (7.2.50)$$

Therefore, a solution of the inverse problem is obtained by solving the linear system

$$(\tilde{\Theta})^T \tilde{\Theta} \hat{\mathbf{w}}^k = (\tilde{\Theta})^T \left(\hat{\mathbf{T}}^k + \Theta_d \hat{\mathbf{w}}^{k-1} - \mathbf{T}_{IC}^k(\tau^k) \right). \quad (7.2.51)$$

As in the piecewise constant case, the system matrix, $(\tilde{\Theta})^T \tilde{\Theta}$, is k -independent.

The proposed methodology for the solution of the Inverse Problem 7.2 is summarized in Algorithm 4. Similarly to the piecewise constant case, for each time interval $(\tau^{k-1}, \tau^k]$, T_{ϕ_i} , T_{d_i} and the related vectors do not change but T_{IC}^k does because its initial condition depends on the temperature field at time τ^{k-1} .

Also in this setting, (7.2.51) is an affine map from the observations, $\hat{\mathbf{T}}^k$, to the heat flux weights, $\hat{\mathbf{w}}^k$. Consequently, we have that the existence and uniqueness of the solution of the inverse problem depends on the invertibility of the matrix $(\tilde{\Theta})^T \tilde{\Theta}$. It is symmetric and positive semi-definite. However, we cannot ensure that it is invertible. In fact, the invertibility depends on the choice of the basis functions, the computational domain and the BCs.

We notice that, also in the piecewise linear case, the offline-online decomposition holds. Moreover, the linear system dimensions are the same of the piecewise constant case and the linearity of the direct problem is still a necessary condition also for Theorem 7.2.

7.3 INVERSE SOLVER FOR S_2^k

In this section, we discuss the solution of the inverse Problem 7.3. In particular, we extend the previously developed methodologies adapting them to the cost function S_2^k as defined in (7.1.8).

Algorithm 4 Inverse solver for the solution of Problem 7.2 with piecewise linear parameterization in time of the heat flux, g .

OFFLINE

Input RBF shape parameter, η ; thermocouples measurement points and times, Ψ, Υ

- 1: Setup RBF parameterization by (3.1.24)
- 2: Compute T_{ϕ_i} for $i = 1, 2, \dots, P$ by solving Problem 7.5
- 3: Compute T_{d_i} for $i = 1, 2, \dots, P$ by solving Problem 7.7
- 4: Assemble matrices $\tilde{\Theta}$ and Θ_d

ONLINE

Input Initial condition, T_0

- 1: Set $k = 1$
 - 2: **while** $k \leq P_t$ **do**
 - 3: Read the thermocouples measurements, $\hat{\mathbf{T}}^k$
 - 4: Compute T_{IC}^k by solving Problem 7.6
 - 5: Assemble \mathbf{T}_{IC}^k
 - 6: Compute $\hat{\mathbf{w}}^k$ by solving (7.2.51)
 - 7: Compute $g_i^k(t)$ by (7.2.4)
 - 8: Compute the heat flux $\mathbf{g}(\mathbf{x}, t)$ for $t \in (\tau^{k-1}, \tau^k]$ by (7.2.1) and (7.2.4)
 - 9: Use (7.2.39) to compute $T_s^k[\mathbf{g}^k]$
 - 10: $k = k + 1$
 - 11: **end while**
-

As shown in detail in Section 9.2, the piecewise linear inverse solver of Section 7.2.2 presents instability issues, under certain conditions. Then, we stated this second inverse problem with the purpose of stabilizing the solution. To do this, we designed Problem 7.3 from Problem 7.2 by adding to the cost function (7.1.8) a term that penalize the heat flux norm $\langle g^k(\tau^k), g^k(\tau^k) \rangle_{L^2(\Gamma_{s_{in}})}$.

Also in this case, we exploit the parameterization of the heat flux (7.2.1). As a consequence, we introduce the inverse problem in terms of \mathbf{w}^k as

Problem 7.8. (Inverse) Given the temperature measurements $\hat{T}(\Psi, \Upsilon)$, find $\hat{\mathbf{w}}^k \in$

\mathbb{R}^P , $1 \leq k \leq P_t$, which minimizes the functional

$$S_2^k[\mathbf{w}^k] = \frac{1}{2} \sum_{i=1}^P [T_s^k[\mathbf{w}^k](\mathbf{x}_i, \tau^k) - \hat{T}^k(\mathbf{x}_i)]^2 + p_g \int_{\Gamma_{s_{in}}} \left(\sum_{j=1}^P w_j^k \phi_j(\mathbf{x}) \right) \left(\sum_{q=1}^P w_q^k \phi_q(\mathbf{x}) \right) d\Gamma. \quad (7.3.1)$$

Notice that (7.3.1) holds true for both the piecewise constant and linear parameterization of the heat flux since

$$g_i^k(\tau^k) = w_i^k, \quad (7.3.2)$$

for (7.2.2) as well as for (7.2.4).

Considering the second term of the right hand side of (7.3.1), we can write

$$\begin{aligned} & \int_{\Gamma_{s_{in}}} \left(\sum_{j=1}^P w_j^k \phi_j(\mathbf{x}) \right) \left(\sum_{q=1}^P w_q^k \phi_q(\mathbf{x}) \right) d\Gamma = \\ & \int_{\Gamma_{s_{in}}} \sum_{j=1}^P \left[w_j^k \phi_j(\mathbf{x}) \left(\sum_{q=1}^P w_q^k \phi_q(\mathbf{x}) \right) \right] d\Gamma = \\ & \sum_{j=1}^P \sum_{q=1}^P \int_{\Gamma_{s_{in}}} w_j^k \phi_j(\mathbf{x}) w_q^k \phi_q(\mathbf{x}) d\Gamma = \\ & \sum_{j=1}^P \sum_{q=1}^P w_j^k w_q^k \int_{\Gamma_{s_{in}}} \phi_j(\mathbf{x}) \phi_q(\mathbf{x}) d\Gamma. \end{aligned} \quad (7.3.3)$$

Let us define the vectors of \mathbb{R}^{P^2}

$$\boldsymbol{\phi}_\phi = \left[\int_{\Gamma_{s_{in}}} \phi_1(\mathbf{x}) \phi_1(\mathbf{x}) d\Gamma \quad \int_{\Gamma_{s_{in}}} \phi_1(\mathbf{x}) \phi_2(\mathbf{x}) d\Gamma \quad \cdots \quad \int_{\Gamma_{s_{in}}} \phi_P(\mathbf{x}) \phi_P(\mathbf{x}) d\Gamma \right]^T, \quad (7.3.4)$$

and

$$\mathbf{a}_w^k = \left[w_1^{k^2} \quad w_1^k w_2^k \quad \cdots \quad w_P^{k^2} \right]^T. \quad (7.3.5)$$

We can now rewrite (7.3.3) as

$$\sum_{j=1}^P \sum_{q=1}^P w_j^k w_q^k \int_{\Gamma_{s_{in}}} \phi_j(\mathbf{x}) \phi_q(\mathbf{x}) d\Gamma = \boldsymbol{\phi}_\phi^T \mathbf{a}_w^k. \quad (7.3.6)$$

Furthermore, deriving (7.3.6) with respect to the weights, we obtain

$$\frac{\partial \left[\int_{\Gamma_{sin}} \left(\sum_{j=1}^P w_j^k \phi_j(\mathbf{x}) \right) \left(\sum_{q=1}^P w_q^k \phi_q(\mathbf{x}) \right) d\Gamma \right]}{\partial w_j^k} = \frac{\partial \phi_\phi^T \mathbf{a}_w^k}{\partial w_j^k} = \phi_\phi^T \frac{\partial \mathbf{a}_w^k}{\partial w_j^k}, \quad \text{for } j = 1, 2, \dots, P. \quad (7.3.7)$$

Considering the case $j = 1$, we have

$$\phi_\phi^T \frac{\partial \mathbf{a}_w^k}{\partial w_1^k} = \phi_\phi^T \begin{bmatrix} 2w_1^k \\ w_2^k \\ \vdots \\ w_P^k \\ w_2^k \\ 0 \\ \vdots \\ 0 \\ w_3^k \\ 0 \\ \vdots \\ 0 \\ w_P^k \\ 0 \\ \vdots \end{bmatrix}, \quad (7.3.8)$$

that we can rewrite as

$$\phi_\phi^T \frac{\partial \mathbf{a}_w^k}{\partial w_1^k} = 2\phi_{\phi_1} w_1^k + \phi_{\phi_2} w_2^k + \phi_{\phi_3} w_3^k + \dots + \phi_{\phi_P} w_P^k + \phi_{\phi_{P+1}} w_2^k + \phi_{\phi_{2P+1}} w_3^k + \dots + \phi_{\phi_{(P-1)P+1}} w_P^k. \quad (7.3.9)$$

Now, by noticing that

$$\phi_{\phi_{(r-1)P+s}} = \int_{\Gamma_{sin}} \phi_r(\mathbf{x}) \phi_s(\mathbf{x}) d\Gamma = \phi_{\phi_{(s-1)P+r}}, \quad 1 \leq r, s \leq P, \quad (7.3.10)$$

we obtain

$$\phi_\phi^T \frac{\partial \mathbf{a}_w^k}{\partial w_1^k} = 2\phi_{\phi_1} w_1^k + 2\phi_{\phi_2} w_2^k + 2\phi_{\phi_3} w_3^k + \dots + 2\phi_{\phi_P} w_P^k = 2\phi_{\phi_{1:P}}^T \mathbf{w}^k. \quad (7.3.11)$$

Similarly, if we consider the general case, we have

$$\begin{aligned} \boldsymbol{\phi}_\phi^T \frac{\partial \mathbf{a}_w^k}{\partial w_j^k} &= 2\boldsymbol{\phi}_{\phi_{(j-1)P+1}} w_1^k + 2\boldsymbol{\phi}_{\phi_{(j-1)P+2}} w_2^k + \\ &2\boldsymbol{\phi}_{\phi_{(j-1)P+3}} w_3^k + \cdots + 2\boldsymbol{\phi}_{\phi_{(j-1)P+P}} w_P^k = \\ &2\boldsymbol{\phi}_{\phi_{(j-1)P+1:(j-1)P+P}}^T \mathbf{w}^k, \quad \text{for } j = 1, 2, \dots, P. \end{aligned} \quad (7.3.12)$$

Therefore, thanks to (7.3.7) and (7.3.12), we can write

$$\begin{aligned} \frac{\partial \left[\int_{\Gamma_{sin}} \left(\sum_{j=1}^P w_j^k \phi_j(\mathbf{x}) \right) \left(\sum_{q=1}^P w_q^k \phi_q(\mathbf{x}) \right) d\Gamma \right]}{\partial w_j^k} &= \\ \boldsymbol{\phi}_\phi^T \frac{\partial \mathbf{a}_w^k}{\partial w_j^k} &= 2\boldsymbol{\phi}_{\phi_{(j-1)P+1:(j-1)P+P}}^T \mathbf{w}^k, \quad \text{for } j = 1, 2, \dots, P. \end{aligned} \quad (7.3.13)$$

Let us define the matrix $\Phi \in \mathbb{M}^{P \times P}$ such that

$$\Phi_{rs} := \int_{\Gamma_{sin}} \phi_r(\mathbf{x}) \phi_s(\mathbf{x}) d\Gamma. \quad (7.3.14)$$

If we now consider the minimization of S_2^k with respect to the weights, \mathbf{w}^k , as in (7.2.9), we have

$$\begin{aligned} \frac{\partial S_2^k[\hat{\mathbf{w}}^k]}{\partial w_j^k} &= \sum_{i=1}^P (R^k[\hat{\mathbf{w}}^k])_i \frac{\partial (\mathbf{T}_s^k[\hat{\mathbf{w}}^k])_i}{\partial w_j^k} + \\ &2p_g \boldsymbol{\phi}_{\phi_{(j-1)P+1:(j-1)P+P}}^T \mathbf{w}^k = 0, \quad \text{for } j = 1, 2, \dots, P. \end{aligned} \quad (7.3.15)$$

Considering the piecewise constant case, thanks to (7.2.31) and (7.3.7), we rewrite (7.3.15) as

$$\Theta^T (\Theta \hat{\mathbf{w}}^k + \mathbf{T}_{IC}^k - \hat{\mathbf{T}}^k) + 2p_g \Phi \hat{\mathbf{w}}^k = \mathbf{0}, \quad (7.3.16)$$

being Θ the matrix defined in (7.2.29). Therefore, for each k , $1 \leq k \leq P_t$, the solution of the inverse problem, $\hat{\mathbf{w}}^k$, is obtained by solving the linear system

$$(\Theta^T \Theta + 2p_g \Phi) \hat{\mathbf{w}}^k = \Theta^T (\hat{\mathbf{T}}^k - \mathbf{T}_{IC}^k). \quad (7.3.17)$$

Similarly, for the piecewise linear case, thanks to (7.2.49) and (7.3.7), we rewrite (7.3.15) as

$$(\tilde{\Theta})^T \left(\tilde{\Theta} \hat{\mathbf{w}}^k - \Theta_d \hat{\mathbf{w}}^{k-1} + \mathbf{T}_{IC}^k - \hat{\mathbf{T}}^k \right) + 2p_g \Phi \hat{\mathbf{w}}^k = \mathbf{0}, \quad (7.3.18)$$

being $\tilde{\Theta}$ and Θ_d the matrices defined in (7.2.48). Therefore, for each k , $1 \leq k \leq P_t$, a solution of the inverse problem, $\hat{\mathbf{w}}^k$, is obtained by solving the linear system

$$\left(\tilde{\Theta}^T \tilde{\Theta} + 2p_g \Phi\right) \hat{\mathbf{w}}^k = \tilde{\Theta}^T (\hat{\mathbf{T}}^k + \Theta_d \hat{\mathbf{w}}^{k-1} - \mathbf{T}_{IC}^k). \quad (7.3.19)$$

The resulting inverse solvers are straightforward modifications of Algorithm 3 and 4. Then, we show them in the following Algorithm 5 and 6.

Algorithm 5 Inverse solver for the solution of Problem 7.3 with piecewise constant parameterization in time of the heat flux, g .

OFFLINE

Input RBF shape parameter, η ; thermocouples measurement points and times, Ψ, Υ ; cost functional parameter, p_g

- 1: Setup RBF parameterization by (3.1.24)
- 2: Compute T_{ϕ_i} for $i = 1, 2, \dots, P$ by solving Problem 7.5
- 3: Assemble matrix Θ by (7.2.29)
- 4: Assemble matrix Φ by (7.3.14)

ONLINE

Input Initial condition, T_0

- 1: Set $k = 1$
 - 2: **while** $k \leq P_t$ **do**
 - 3: Read the thermocouples measurements, $\hat{\mathbf{T}}^k$
 - 4: Compute T_{IC}^k by solving Problem 7.6
 - 5: Assemble \mathbf{T}_{IC}^k
 - 6: Compute $\hat{\mathbf{w}}^k$ by solving (7.3.17)
 - 7: Compute $g_i^k(t)$ by (7.2.2)
 - 8: Compute the heat flux $\mathbf{g}(\mathbf{x}, t)$ for $t \in (\tau^{k-1}, \tau^k]$ by (7.2.1) and (7.2.2)
 - 9: Use (7.2.21) to compute $T_s^k[\mathbf{g}^k]$
 - 10: $k = k + 1$
 - 11: **end while**
-

As a final remark, notice that for $p_g = 0 \frac{K^2}{W^2}$, we end up with the same solution as for S_1^k .

7.4 REGULARIZATION

After the development of novel inverse solvers, we provide a brief discussion about regularization. It is well known that inverse problems as the ones here considered are

Algorithm 6 Inverse solver for the solution of Problem 7.3 with piecewise linear parameterization in time of the heat flux, g .

OFFLINE

Input RBF shape parameter, η ; thermocouples measurement points and times, Ψ, Υ ; cost functional parameter, p_g

- 1: Setup RBF parameterization by (3.1.24)
- 2: Compute T_{ϕ_i} for $i = 1, 2, \dots, P$ by solving Problem 7.5
- 3: Compute T_{d_i} for $i = 1, 2, \dots, P$ by solving Problem 7.7
- 4: Assemble matrices $\tilde{\Theta}$ and Θ_d
- 5: Assemble matrix Φ by (7.3.14)

ONLINE

Input Initial condition, T_0

- 1: Set $k = 1$
 - 2: **while** $k \leq P_t$ **do**
 - 3: Read the thermocouples measurements, $\hat{\mathbf{T}}^k$
 - 4: Compute T_{IC}^k by solving Problem 7.6
 - 5: Assemble \mathbf{T}_{IC}^k
 - 6: Compute $\hat{\mathbf{w}}^k$ by solving (7.3.19)
 - 7: Compute $g_i^k(t)$ by (7.2.4)
 - 8: Compute the heat flux $\mathbf{g}(\mathbf{x}, t)$ for $t \in (\tau^{k-1}, \tau^k]$ by (7.2.1) and (7.2.4)
 - 9: Use (7.2.39) to compute $T_s^k[\mathbf{g}^k]$
 - 10: $k = k + 1$
 - 11: **end while**
-

ill-posed. This means that for our problem at least one of the following properties does not hold: for all admissible data, a solution exists; for all admissible data, the solution is unique; the solution depends continuously on the data [49]. In our discussion, we turned the infinite dimensional inverse Problem 7.2 into the solution of the discrete linear systems (7.2.33) and (7.2.51) by making some assumptions on the heat flux (i.e. parameterizing it). In this new setting, if the matrices $\Theta^T \Theta$ and $\tilde{\Theta}^T \tilde{\Theta}$ are invertible, we have the existence of a unique solution for our inverse problem.

As we will see in the numerical tests section, it turns out that these matrices are very ill-conditioned. This can cause the matrix to be numerically rank deficient, losing the uniqueness of a solution. However, this is not the only concern. We still have the problem of a continuous dependence of the solution on the data. The

ill-conditioning of the linear system causes that, if we have some noise in the data vector (as usual in an industrial measurement equipment), the solution of the linear system diverges from the correct value.

To address both these problems, we require regularization. There are several techniques available for regularizing a discrete ill-posed problem as the present one. In general, they are divided into direct methods like TSVD and Tikhonov regularization, and iterative methods such as the CGM. For a deep discussion of all regularization methods, we refer the interested reader to Hansen's monograph on the subject [65].

In the present investigation, we use again the TSVD as described in Section 3.1.2. Together with the classical aforementioned regularization method, we investigate also the regularization by discretization [4, 80]. Using this method, we exploit the regularizing properties of coarsening the time and/or space discretization to improve the heat flux estimation. In Chapter 9, we will test the performance of these regularization methods also by adding noise to the thermocouples measurements.

7.5 DISCRETIZATION SELECTION ALGORITHM

To conclude this chapter, we propose an algorithm for the automated selection of some of the parameters required by Algorithm 6. As shown in Chapter 9, the numerical tests highlight that this algorithm is very sensitive to the mesh and time discretization refinement as well as to the parameter p_g . We anticipate here that this inverse solver shows severe instabilities for fine discretizations. However, these instabilities are effectively eliminated for values of p_g that are above a threshold that depends on the discretization refinement. In for these values of p_g , we notice a drastic decrease of the dependency of the algorithm from the discretization.

However, the uncontrolled increase in p_g does not lead to a monotonic improvement of the inverse solver performances. As can be observed in the numerical results of Chapter 9 (see Figures 9.2.7, 9.2.8, 9.2.9, 9.3.7, 9.3.8), the dependency of the algorithm from p_g is such that it is unstable for low values of p_g then, increasing further p_g , it sharply achieves an optimum of performance before reaching a plateau at which the algorithm is stable but the term $\langle g^k(\tau^k), g^k(\tau^k) \rangle_{L^2(\Gamma_{sin})}$ in (7.1.8) overcomes the measurements distance one, S_1^k defined in (7.1.7). Thus, for too high values of p_g , we have a stable algorithm that is almost independent from the discretization refinement but that provides poor heat flux estimations.

To allow an industrial use of the proposed inverse solver, the objective of this section is to develop a method for automatically selecting the Δt , the mesh and the value of p_g such that the algorithm is stable and accurately estimates the mold-steel heat flux.

In developing such method, we assume to have available a reliable dataset of thermocouples measurements, $\hat{T}_{train}(\Psi, \Upsilon_{train})$, that we can use to perform this tuning offline. Moreover, we assume that, independently from the mold physical parameters and the heat flux values, this inverse solver always shows the previously described behaviour with respect to p_g . In particular, we assume that, for values of p_g higher than a problem specific threshold, the algorithm is stable for all the discretizations and independent from them (i.e. we obtain similar solutions for any given mesh and Δt).

We recall, that in the real industrial case, we do not have any information about the true heat flux that we want to estimate. Thus, this selection methodology cannot be based on the heat flux estimation error. However, Figures 9.2.10 and 9.3.9 show that the measurement discrepancy functional S_1^k and the heat flux estimation error have a similar behaviour as functions of p_g and we will use this quantity to determine the quality of the heat flux estimation.

All that said, we begin by selecting an ordered set of meshes $(\Delta x_1, \Delta x_2, \dots, \Delta x_{n_M})$ and an ordered set of timestep sizes $(\Delta t_1, \Delta t_2, \dots, \Delta t_{n_t})$. We order them from the finest to the coarsest discretization (i.e. $\Delta x_1 < \Delta x_2 < \dots < \Delta x_{n_M}$ and $\Delta t_1 < \Delta t_2 < \dots < \Delta t_{n_t}$). Then, our first objective is to identify a p_g^0 within the aforementioned stability region.

To do it, we start with a tentative p_g^0 . For this value of the parameter, we solve the inverse problem on the training measurement dataset for all Δx and Δt . Let us denote by $T[\Delta x, \Delta t]$ the corresponding solution. Having done so, we compute

$$\Delta T := \max_{i,j,q,p} \|T[\Delta x_i, \Delta t_j] - T[\Delta x_q, \Delta t_p]\|_{L^\infty((0,t_f]; L^2(\Omega_s))}. \quad (7.5.1)$$

If we have

$$\Delta T > \Delta x_{n_M} + \Delta t_{n_t}, \quad (7.5.2)$$

we consider that the solution is too dependent on the discretization refinement. Then, we increase the value of p_g^0 and redo the calculations until

$$\Delta T \leq \Delta x_{n_M} + \Delta t_{n_t}, \quad (7.5.3)$$

is satisfied.

Once we find a value of p_g^0 within the stability region, we choose the discretization setup $(\Delta x^1, \Delta t^1)$ that corresponds to the minimum for p_g^0 of

$$m_S[\Delta x, \Delta t, p_g] := \text{mean}_k \left(S_1^k[\Delta x, \Delta t, p_g] \right). \quad (7.5.4)$$

Once we select Δx^1 and Δt^1 , we choose p_g^1 as the value of p_g that minimizes $m_S[\Delta x^1, \Delta t^1, p_g]$. Then, we fix $p_g = p_g^1$ and we solve again the inverse problem for all the considered discretization setups. If the previously selected discretization is the one that corresponds to the lowest value of $m_S[\Delta x, \Delta t, p_g^1]$, we choose Δx^1 , Δt^1 , and $p_g = p_g^1$, and we stop the process. Otherwise, we continue iterating by selecting Δx^2 and Δt^2 as the ones corresponding to the smallest $m_S[\Delta x, \Delta t, p_g^1]$ and looking for the p_g^2 that minimizes $m_S[\Delta x^2, \Delta t^2, p_g]$, and so on. We summarize all this process in Algorithm 7.

This method allows a data-driven, automated selection of the discretization refinement and the p_g parameter. This result comes to the cost of computing $n_M \cdot n_t$ solutions to the inverse problem at each iteration. If the available memory allows it, we can keep in the memory the results of the offline computations related to each discretization. Otherwise, we have to recompute every time these offline phases. However, we designed this algorithm to be used offline. Then, even if it is computationally expensive, we can run it before the caster starts to work and it only requires the dataset of thermocouples measurements \hat{T}_{train} .

Algorithm 7 Offline selection of the mesh, the timestep size and p_g for the inverse solver in Algorithm 6.

Input Ordered set of meshes, $(\Delta x_1, \Delta x_2, \dots, \Delta x_{n_M})$; ordered set of timestep sizes, $(\Delta t_1, \Delta t_2, \dots, \Delta t_{n_t})$; p_g^0 ; training set, $\hat{T}_{train}(\Psi, \Upsilon_{train})$

- 1: **while** $\Delta T > \Delta x_{n_M} + \Delta t_{n_t}$ **do** ▷ Identify stability region
- 2: **for** $i = 1$ to n_M **do**
- 3: **for** $j = 1$ to n_t **do**
- 4: Solve the inverse problem on the training set, \hat{T}_{train} , by using Algorithm 6 with Δx_i , Δt_j and $p_g = p_g^0$
- 5: Compute $m_S[\Delta x_i, \Delta t_j, p_g^0]$ by (7.5.4)
- 6: **end for**
- 7: **end for**
- 8: Compute ΔT by (7.5.3)
- 9: **if** $\Delta T > \Delta x_{n_M} + \Delta t_{n_t}$ **then**
- 10: $p_g^0 = 10p_g^0$
- 11: **end if**
- 12: **end while**
- 13: Choose Δx^0 and Δt^0 corresponding to $\min_{i,j} m_S[\Delta x_i, \Delta t_j, p_g^0]$ for $1 \leq i \leq n_M, 1 \leq j \leq n_t$
- 14: $l = 1, f = 0$
- 15: **while** $f = 0$ **do**
- 16: Find $p_g^l = \operatorname{argmin}_{p_g} (m_S[\Delta x^{l-1}, \Delta t^{l-1}, p_g])$
- 17: **for** $i = 1$ to n_M **do**
- 18: **for** $j = 1$ to n_t **do**
- 19: Solve the inverse problem on the training set, \hat{T}_{train} , by using Algorithm 6 with Δx_i , Δt_j and $p_g = p_g^l$
- 20: Compute $m_S[\Delta x_i, \Delta t_j, p_g^l]$ by (7.5.4)
- 21: **end for**
- 22: **end for**
- 23: Choose Δx^l and Δt^l corresponding to $\min_{i,j} m_S[\Delta x_i, \Delta t_j, p_g^l]$ for $1 \leq i \leq n_M, 1 \leq j \leq n_t$
- 24: **if** $\Delta x^l = \Delta x^{l-1}$ & $\Delta t^l = \Delta t^{l-1}$ **then**
- 25: $f = 1$
- 26: **end if**
- 27: $l = l + 1$
- 28: **end while**

8 MODEL ORDER REDUCTION FOR THE SPEED-UP OF INVERSE SOLVERS

In the present chapter, we exploit MOR techniques to reduce the computational cost of the inverse solvers that we developed in the previous chapter. After a brief introduction and a literature survey on MOR, we discuss in Section 8.2 the methodology that we use for the creation of the Reduced Order Model (ROM) that is required for the real-time solution of the considered inverse problems. Then, Section 8.3 introduces the incremental approach that we exploit to develop an automated algorithm for the construction of the surrogate model. Finally, we conclude this chapter with Section 8.4 where we implement these MOR techniques in a novel inverse solver.

8.1 STATE OF THE ART

In many fields of application, we have a parameterized model that implicitly connects some input parameters to a quantity of interest. Examples of input parameters can be the angle of attack of a wing or the thermal conductivity of a medium with the possible output being the wing drag and the temperature field, respectively.

The accurate computation of the solutions to these problems is clearly of broad interest. However, MOR focuses on the efficiency of this computation for problems in which the solution is sought for a large number of different parameter values, with a limited computational power or in a small CPU time (e.g. real-time).

If we focus on continuous mechanics only, we are usually dealing with Parameterized Partial Differential Equation (PPDE) models. In this framework, several MOR techniques have been studied in the literature (Reduced Basis (RB) [68], NN [19] or Krylov subspace projection [22], just to name a few). Due to the vastness of the field, we consider in the further course of this chapter the RB MOR only.

This technique is well established for the reduction of PPDEs models (see e.g. [12, 13, 40, 68, 126]). As such, it has been applied into several fields, ranging from fluid dynamics (see [86, 105, 144]), heat transfer (see [21, 41, 75]), structural dynamics (see [55, 106]), electromagnetism (see [135]) or neuroscience (see [40]).

Once these ROMs have been developed and tested, researchers start to exploit them to explore the parameter space of the PPDEs in multi query problems with a limited computational cost. In this regard, MOR literally opened the doors to solve

efficiently these problems that require several simulations to obtain a solution. In fact, we find in the recent literature several application of RB MOR to, e.g., optimization (see e.g. [14, 15, 57, 132, 136, 173]), bifurcation (see [116–118]), uncertainty quantification (see [16, 53, 59]), but also multiphysics and multiscale problems such as fluid-structure interaction (see [109, 167]), aereoacoustic (see [71, 140]) or bone structural analysis (see [66, 171]), just to name a few.

Between these previous attempts, it is interesting for the present research to mention those that are related to optimal control and inverse problems. In particular, Negri et al. [103] proposed a certified RB for the rapid and reliable solution of parameterized optimal control problems governed by elliptic Partial Differential Equation (PDE). Using this framework, Strazzullo et al. [148–150] moved a step forward by developing RB MOR for the solution of nonlinear and time-dependent parametric optimal flow control problems. We also mention the work of Bernreuther et al. [18] who investigated the use of RB for the optimal control of non-smooth semilinear elliptic PDEs.

If we now consider the present problem, we have that the proposed Algorithms 3 and 4 require, for the solution of the inverse problem, the computation of the FOM Problem 7.6 for each measurement instant k . Being this a FOM, the computational cost of its solution depends on the size of the space and time discretization. Hence, we cannot ensure the real-time performance of the inverse solvers. Then, our objective is to develop a ROM of Problem 7.6 to reduce the computational cost of its solution.

Among all available techniques to develop a ROM for Problem 7.6, we apply the RB Proper Orthogonal Decomposition (POD)-Galerkin method (see [143]). As we discuss in detail in Section 8.2, it consists in sampling the parameter space, computing the related snapshots, and, eventually, perform a POD on these snapshots to obtain the RB. Finally, we obtain the ROM by performing a Galerkin projection of the FOM equations onto the RB space.

In our setting, we cannot apply this technique straightforwardly. In fact, since our parameter (the IC $T_s^{k-1}(\cdot, \tau^{k-1})$) can be any of the possible solutions to Problem 7.1, we are not able to sample a priori the parameter space as usual with these MOR techniques. For this reason, in Section 8.3, we combine the classical RB POD-Galerkin with the incremental POD technique presented in [54] to create a data-driven sampling algorithm that progressively enrich the RB space. Consequently, the creation of the RB space requires a second offline phase in which we input a dataset of thermocouples measurements to setup the RB space.

We begin this chapter by describing the RB POD-Galerkin MOR method that we use to develop a surrogate model of Problem 7.6. Then, in Section 8.3, we develop an

incremental approach for the automated creation of the surrogate model. To conclude, we present in Section 8.4 the novel inverse solvers that implement the aforementioned MOR techniques.

8.2 RB POD-GALERKIN REDUCED ORDER MODEL

Considering Problem 7.6, its input parameter is the IC, $T_s^{k-1}(\cdot, \tau^{k-1})$. Then, the required surrogate model should be such that, given $T_s^{k-1}(\cdot, \tau^{k-1})$, it provides an approached solution, \tilde{T}_{IC}^k , to Problem 7.6 in $(\tau^{k-1}, \tau^k]$. To simplify the notation, we denote here the IC, $T_s^{k-1}(\cdot, \tau^{k-1})$, by T_0^k .

In the present section, we introduce the general methodology for the creation of the aforementioned surrogate model. Firstly, we discuss the typical procedure for the computation of the POD RB space. Secondly, we perform a Galerkin projection of the FOM discrete equations on this space to obtain the ROM.

8.2.1 Reduced Basis Space

We now discuss the general approach for the computation of the RB. Notice that we present it to provide the reader with a complete discussion on the subject even if in this work we use the incremental POD technique described in Section 8.3 for the computation of the RB.

To simplify the notation and just in this subsection, we drop the index k and consider the interval $(\tau^0, \tau^1]$ only. This simplification does not reduce the generality of the discussion since the only dependence on k of Problem 7.6 is in the IC. Then, we can think of solving the problem in $(\tau^0, \tau^1]$ for a general T_0^1 , and translate it to the k -th interval as we did for Problem 7.5 in (7.2.20).

Assuming for now to be able to select a set $\mathcal{K} = \{T_{0_1}, \dots, T_{0_{N_p}}\} \subset H^1(\Omega)$ of N_p samples for the parameter T_0 that properly samples the parameter space $H^1(\Omega)$, we start by solving Problem 7.6 for each $T_{0_p} \in \mathcal{K}$ in $(\tau^0, \tau^1]$. Then, for each $1 \leq p \leq N_p$, we obtain the solutions, $\mathbf{T}_{IC}[T_{0_p}](t^n)$, for $1 \leq n \leq N_t$, where $\mathbf{T}_{IC}[T_{0_p}](t^n) \in \mathbb{R}^{N_h}$ is the FOM discrete solution to Problem 7.6 in the interval $(\tau^0, \tau^1]$ with IC $T_0 = T_{0_p}$, at time t^n , N_h being the cardinality of the mesh. As usual in the MOR community, we call these solutions ‘‘snapshots’’. We denote their number by $N_s = N_p \times N_t$ and collect them into the matrix $S \in \mathbb{M}^{N_h \times N_s}$, such that

$$S_{i, n+N_t(p-1)} := (\mathbf{T}_{IC})_i[T_{0_p}](t^n), \quad \text{for } 1 \leq i \leq N_h, 1 \leq p \leq N_p, 1 \leq n \leq N_t. \quad (8.2.1)$$

To generate the RB space from the snapshots, several techniques have been con-

sidered in literature (such as POD, Proper Generalized Decomposition (PGD) or the greedy sampling strategy, for more details the reader may see [34, 125, 131]). In this work, we exploit a POD approach applying the POD onto the full snapshots matrix that includes both the time and parameter dependency. In time dependent problems, other approaches were also considered in the literature. For example, one can use the POD-Greedy approach [64] or the nested POD approach where the POD is applied before in the time domain and then on the parameter space [60].

Given the function $\mathbf{T}_{IC}(t^n) : (\tau^0, \tau^1] \rightarrow \mathbb{R}^{N_h}$, with the realizations $\mathbf{T}_{IC_1}, \mathbf{T}_{IC_2}, \dots, \mathbf{T}_{IC_{N_s}}$, the objective of the POD problem is to find the coefficients $a_{(1,1)}, \dots, a_{(1,N_s)}, \dots, a_{(N_s,1)}, \dots, a_{(N_s,N_s)}$ and the functions $\varphi_1, \dots, \varphi_{N_s} \in \mathbb{R}^{N_h}$ that minimize the quantity

$$E_q := \sum_{l=1}^{N_s} \left\| \mathbf{T}_{IC_l} - \sum_{p=1}^q a_{(l,p)} \varphi_p \right\|, \quad \text{for } q = 1, \dots, N_s, \quad (8.2.2)$$

with $\varphi_l^T M \varphi_m = \delta_{lm}$ for all $l, m = 1, \dots, N_s$, where M is the diagonal mass matrix defined in (6.1.20), and $\mathbf{T}_{IC_l} = \mathbf{T}_{IC}[T_{0_p}](t^n)$, being $p - 1$ the integer quotient of l/N_t and n its remainder.

In [84] it is shown that this minimization problem is equivalent to solving the eigenvalue problem

$$CQ = Q\lambda, \quad (8.2.3)$$

where $C_{lm} := \mathbf{T}_{IC_l}^T M \mathbf{T}_{IC_m}$ for $l, m = 1, \dots, N_s$ is the snapshots correlation matrix, Q is a square matrix with the eigenvectors as columns and λ is a vector of eigenvalues. Then, the basis functions can be obtained by (see [143])

$$\varphi_l = \frac{1}{N_s \lambda_l} \sum_{m=1}^{N_s} Q_{lm} \mathbf{T}_{IC_m}, \quad \text{for } 1 \leq l \leq N_s. \quad (8.2.4)$$

This is the so called method of snapshots introduced by Sirovich [141] in 1987. However, the basis (8.2.4) can also be found by performing a SVD of the matrix (see [58])

$$S_M := M^{1/2} S = U \Sigma W^T, \quad (8.2.5)$$

where $M^{1/2}$ is the coefficient-wise square root of the mass matrix, $U \in \mathbb{M}^{N_h \times N_h}$ and $W \in \mathbb{M}^{N_s \times N_s}$ are orthogonal matrices, and $\Sigma \in \mathbb{M}^{N_h \times N_s}$ is the diagonal matrix with the singular values of S_M on the diagonal in a decreasing order. The matrices U and W contain on the columns the orthogonal vectors to the $S_M S_M^T$ and $S_M^T S_M$ matrices, respectively. In this setting, the POD basis are the columns of U .

In POD applications, it is typical to only need information about singular vectors corresponding to nonzero singular values. Let $r_S = \text{rank}(S_M) \leq \min(N_s, N_h)$, i.e. S_M has exactly r_S positive singular values, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{r_S}$. Then, defining

$$\hat{U} := (U_{lm})_{\substack{1 \leq l \leq N_h \\ 1 \leq m \leq r_S}} \in \mathbb{M}^{N_h \times N_{r_S}}, \quad (8.2.6)$$

and the diagonal matrix $M^\dagger \in \mathbb{M}^{N_h \times N_h}$, such that

$$M_{ii}^\dagger := \frac{1}{\sqrt{M_{ii}}}, \quad (8.2.7)$$

the POD modes matrix is given by

$$M^\dagger \hat{U} = [\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_{r_S}]. \quad (8.2.8)$$

To conclude, we define the RB space as $\text{span}(\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_{N_r})$ where we only keep the first $N_r \leq r_S$ dominant modes. Then, we define the RB matrix as

$$L := [\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_{N_r}]. \quad (8.2.9)$$

8.2.2 Reduced Order Model

In this section, we discuss the creation of the ROM for Problem 7.6 by performing a Galerkin projection of its discrete equation onto the RB space, given the RB POD space and the related matrix L , defined in (8.2.9).

We have that the \mathbf{T}_{IC}^k temperature field can be approximated in the RB space with

$$\mathbf{T}_{IC}^k[T_0^k](\hat{t}^n) \approx \sum_{l=1}^{N_r} (\tilde{\mathbf{T}}_{IC}^k)_l(\hat{t}^n, T_0^k) \boldsymbol{\varphi}_l(\mathbf{x}) = L \tilde{\mathbf{T}}_{IC}^k(\hat{t}^n, T_0^k), \quad (8.2.10)$$

where $\hat{t}^n := t^{(k-1)N_t+n}$ and the reduced coefficients column vector, $\tilde{\mathbf{T}}_{IC}^k \in \mathbb{R}^{N_r}$, depend on the time and initial field only, while the basis functions $\boldsymbol{\varphi}$ depend on the physical space. To derive an equation for the unknown vector of coefficients $\tilde{\mathbf{T}}_{IC}^k$, we discretize Problem 7.6 as for (6.1.26), obtaining

$$(\rho C_p M + \Delta t A) \mathbf{T}_{IC}^k(\hat{t}^{n+1}) = \rho C_p M \mathbf{T}_{IC}^k(\hat{t}^n) + \Delta t \tilde{\mathbf{b}}, \quad n = 0, 1, \dots, N_t - 1. \quad (8.2.11)$$

with

$$(\tilde{\mathbf{b}})_i = \sum_{\sigma \in \mathcal{E}_{K_i} \cap \Gamma_{sf}} q_{K_i, \sigma} T_{f_\sigma}. \quad (8.2.12)$$

Then, by performing a Galerkin projection of (8.2.11) onto the POD RB space and thanks to (8.2.10), we have

$$\begin{aligned} & (\rho C_p L^T M L + \Delta t L^T A L) \tilde{\mathbf{T}}_{IC}^k(\hat{t}^{n+1}) = \\ & \rho C_p L^T M L \tilde{\mathbf{T}}_{IC}^k(\hat{t}^n) + \Delta t L^T \tilde{\mathbf{b}}, \quad n = 0, 1, \dots, N_t - 1. \end{aligned} \quad (8.2.13)$$

Defining

$$M_r := L^T M L, \quad A_r := L^T A L, \quad \tilde{\mathbf{b}}_r := L^T \tilde{\mathbf{b}}, \quad (8.2.14)$$

we rewrite (8.2.13) as

$$(\rho C_p M_r + \Delta t A_r) \tilde{\mathbf{T}}_{IC}^k(\hat{t}^{n+1}) = \rho C_p M_r \tilde{\mathbf{T}}_{IC}^k(\hat{t}^n) + \Delta t \tilde{\mathbf{b}}_r, \quad n = 0, 1, \dots, N_t - 1, \quad (8.2.15)$$

obtaining the ROM for Problem 7.6, where the IC is

$$\tilde{\mathbf{T}}_{IC}^k(\hat{t}^0) = L^{-1} \mathbf{T}_0^k = L^T \mathbf{T}_0^k, \quad (8.2.16)$$

thanks to the orthogonality of L .

So in summary, to compute the ROM of Problem 7.6, we start by generating a set of snapshots of the solution. Then, we perform a POD on the snapshots to compute the RB space. Selecting the number of POD modes to use, we assemble the RB matrix L in (8.2.9). Once this matrix is available, we compute the reduced matrices M_r and A_r , and the reduced vector $\tilde{\mathbf{b}}_r$. All this computations give us the ROM (8.2.15).

Notice that, if we can generate the snapshots a priori, all these computation can be done in an offline phase. Then, during the online phase, we solve the ROM linear system (8.2.15) and, if necessary, we reconstruct the FOM field by (8.2.10). Moreover, we notice that in general we choose $N_r \ll N_h$ (in the order of tens) according to the singular values decay. Thus, the size of (8.2.15) is such that the ROM online phase is dramatically faster than computing a solution of the FOM.

8.3 INCREMENTAL CONSTRUCTION OF THE REDUCED BASIS SPACE

As previously mentioned, to sample a priori the parameter space during a single offline phase is not possible in the present application because T_0^k can be any function of $H^1(\Omega)$. However, we can assume to have available during the offline phase a set

of thermocouples measurements coming from previous runs of the CC mold (we will call it the “training set”). With this set of measurements, we can run the inverse solvers that we designed in Chapter 7. Each iteration of these algorithms would provide a T_0^k . Thus, we can think of using them as proper samples of the parameter of Problem 7.6.

This is a possible procedure for the generation of the required snapshots. However, it would require us to store in the memory all the FOM solution of Problem 7.6 and, at the end, to perform the POD of a huge snapshots matrix. Moreover, we could not ensure that the resulting RB space is a good representation of all the T_0^k that could appear during the online phase.

To overcome these issues, we propose a novel methodology for the construction of the RB space. In particular, we want to exploit the incremental SVD algorithm developed by Fareed et al [54] to create an automated methodology for the generation of the ROM POD basis. It should be such that it does not require the storage of all the snapshots and it can quickly decide whether an enrichment of the RB space is needed or we can perform a ROM computation. Moreover, it should ensure, in the latter case, that the incoming IC is properly represented by the POD basis.

To derive this novel incremental methodology, we first present in Section 8.3.1 the incremental POD algorithm to progressively expand our RB space without requiring the storage of all the snapshots. Then, in Section 8.3.2, we derive a projection error estimator that can quickly estimate the error that we commit when we project the incoming IC, T_0^k , into the RB space. This error estimator is necessary to determine whether to enrich the POD space by performing a FOM solution of Problem 7.6 obtaining the related snapshot, or to use the ROM. Finally, we present in Section 8.3.3 a novel algorithm for the data-driven, automated, incremental construction of the RB space.

8.3.1 Incremental Proper Orthogonal Decomposition Algorithm

Incremental POD algorithms have been proposed in the literature by several authors (see e.g. [20, 113, 115, 172]). In general, the idea is to obtain the SVD of a matrix at which we add a column knowing its previous SVD and the new column vector. Here, the matrix is the snapshots matrix S_M and the new column vector is a new snapshot $\mathbf{T}_{IC_{N_s+1}}$. In the present work, we propose a modification of the algorithm developed by Fareed et al [54] and we consider it in the context of our application.

Let us assume to have the SVD of the snapshots matrix $S_M = U\Sigma W^T$ of dimen-

sion $N_h \times N_s$. Now, we consider to add a new snapshot $\mathbf{T}_{IC_{N_s+1}}$. Our goal is to obtain the SVD of the matrix $[S_M \quad \mathbf{T}_{IC_{N_s+1}}]$ without forming the matrices S_M or $[S_M \quad \mathbf{T}_{IC_{N_s+1}}]$.

Let us define

$$\begin{aligned} \tilde{\mathbf{T}}_{IC_{N_s+1}} &= U^T M \mathbf{T}_{IC_{N_s+1}}, & \mathbf{T}_{IC_{N_s+1}}^{\parallel} &:= U \tilde{\mathbf{T}}_{IC_{N_s+1}}, \\ \mathbf{T}_{IC_{N_s+1}}^{\perp} &:= \mathbf{T}_{IC_{N_s+1}} - \mathbf{T}_{IC_{N_s+1}}^{\parallel}, \end{aligned} \quad (8.3.1)$$

the projection error

$$\epsilon_{\perp} := \left\| \mathbf{T}_{IC_{N_s+1}}^{\perp} \right\|_{L^2(\Omega)}, \quad (8.3.2)$$

$$\mathbf{T}_{IC_{N_s+1}}^{\epsilon} := \frac{\mathbf{T}_{IC_{N_s+1}}^{\perp}}{\epsilon_{\perp}}, \quad (8.3.3)$$

and the relative projection error

$$\epsilon_{\perp rel} := \frac{\epsilon_{\perp}}{\left\| \mathbf{T}_{IC_{N_s+1}} \right\|_{L^2(\Omega)}}. \quad (8.3.4)$$

If $\epsilon_{\perp} > 0$, we have the identity (see [54])

$$\begin{aligned} [S_M \quad \mathbf{T}_{IC_{N_s+1}}] &= [U \Sigma W^T \quad \mathbf{T}_{IC_{N_s+1}}] = \\ &= \begin{bmatrix} U & \mathbf{T}_{IC_{N_s+1}}^{\epsilon} \end{bmatrix} \begin{bmatrix} \Sigma & \tilde{\mathbf{T}}_{IC_{N_s+1}} \\ 0 & \epsilon_{\perp} \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}^T. \end{aligned} \quad (8.3.5)$$

We can find the SVD of $[S_M \quad \mathbf{T}_{IC_{N_s+1}}]$ by computing the SVD of the matrix

$$\tilde{S}_M := \begin{bmatrix} \Sigma & \tilde{\mathbf{T}}_{IC_{N_s+1}} \\ 0 & \epsilon_{\perp} \end{bmatrix} = \tilde{U} \tilde{\Sigma} \tilde{W}^T. \quad (8.3.6)$$

Then, the SVD of $[S_M \quad \mathbf{T}_{IC_{N_s+1}}]$ is given by

$$[S_M \quad \mathbf{T}_{IC_{N_s+1}}] = \left(\begin{bmatrix} U & \mathbf{T}_{IC_{N_s+1}}^{\epsilon} \end{bmatrix} \tilde{U} \right) \tilde{\Sigma} \left(\begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} \tilde{W} \right)^T. \quad (8.3.7)$$

In practice, we perform truncation when $\epsilon_{\perp rel}$ is below a threshold value ϵ_{SVD} . Moreover, since the orthogonality of the matrix U can be lost in this process, we compute the orthogonality parameter,

$$o = U_{(:,N_s+1)}^T M U_{(:,1)}. \quad (8.3.8)$$

In case it is above the threshold value ϵ_{ortho} , we perform a weighted Gram Schmidt orthogonalization as suggested by Fareed et al [54].

Algorithm 8 summarizes the incremental POD procedure. Thanks to it, we are now able to progressively enrich the RB space without recomputing every time all the POD modes and without keeping in memory all the snapshots.

8.3.2 Projection Error Estimator

As anticipated, we now discuss the development of an online error estimator for the projection of an IC, T_0 , onto the RB space.

In general, given the orthogonal component of the projected field

$$\mathbf{T}_0^\perp := \mathbf{T}_0 - LL^T M \mathbf{T}_0, \quad (8.3.9)$$

the projection error can be computed as

$$\epsilon_{proj} := \frac{\|\mathbf{T}_0^\perp\|_{L^2(\Omega)}}{\|\mathbf{T}_0\|_{L^2(\Omega)}}. \quad (8.3.10)$$

However, this is a FOM computation that cannot be done in real-time. Hence, we propose to estimate this quantity by computing it on some accurately selected discrete points only. Then, if the number of such points is reasonably small, this estimation would be computationally cheap.

Let us consider as such estimation points a set $\mathcal{A} := \{\alpha_1, \dots, \alpha_{N_{MP}}\}$ in the interior of the domain, with $N_{MP} < N_r$. Then, an estimation of the projection error (8.3.10) at these points reads

$$\epsilon_{proj} \approx \frac{\|\mathbf{T}_0^\perp\|_{L^2(\mathcal{A})}}{\|\mathbf{T}_0\|_{L^2(\mathcal{A})}}. \quad (8.3.11)$$

To select the points \mathcal{A} that provide a meaningful approximation of ϵ_{proj} , we exploit the procedure typically used in the Discrete Empirical Interpolation Method (DEIM) (see e.g. [28, 29]) for the selection of the “magic points”. The selection process of DEIM algorithm essentially involves minimizing the error of the approximation via the selected index in each iteration.

To describe the process, we introduce the following quantities. We define the matrix $L_l \in \mathbb{M}^{N_h \times l}$ as the matrix containing the first l columns of the POD matrix L , i.e.

$$L_l := [\varphi_1, \varphi_2, \dots, \varphi_l]. \quad (8.3.12)$$

Algorithm 8 Incremental POD

Input Snapshot, $\mathbf{T}_{IC_{N_s+1}} \in \mathbb{R}^{N_h}$; SVD tolerance, ϵ_{SVD} ; Orthogonality tolerance, ϵ_{ortho}

```

1: if  $N_s = 0$  then                                     ▷ Initialize the POD space
2:    $\Sigma = \left[ \sqrt{\mathbf{T}_{IC_1}^T M \mathbf{T}_{IC_1}} \right]$ 
3:    $U = \mathbf{T}_{IC_1} / \sqrt{\mathbf{T}_{IC_1}^T M \mathbf{T}_{IC_1}}$ 
4:    $N_s = 1$ 
5: else
6:   Compute  $\tilde{\mathbf{T}}_{IC_{N_s+1}}$ ,  $\mathbf{T}_{IC_{N_s+1}}^{\parallel}$ , and  $\mathbf{T}_{IC_{N_s+1}}^{\perp}$ , by (8.3.1)
7:   Compute the projection error,  $\epsilon_{\perp}$ , by (8.3.2)
8:   Compute the relative projection error,  $\epsilon_{\perp rel}$ , by (8.3.4)
9:   Compute  $\tilde{S}$  by (8.3.6)
10:  if  $\epsilon_{\perp rel} < \epsilon_{SVD}$  then
11:    Set  $\tilde{S}(N_s + 1, N_s + 1) = 0$ 
12:  end if
13:   $[\tilde{U}, \tilde{\Sigma}, \tilde{W}] = SVD(\tilde{S})$ 
14:  if  $\epsilon_{\perp rel} < \epsilon_{SVD}$  or  $N_s \geq N_h$  then
15:     $U = U\tilde{U}_{(1:N_s, 1:N_s)}$ 
16:  else
17:    Compute  $\mathbf{T}_{IC_{N_s+1}}^{\epsilon}$  by (8.3.3)
18:     $U = \left[ U \quad \mathbf{T}_{IC_{N_s+1}}^{\epsilon} \right] \tilde{U}$ 
19:     $N_s = N_s + 1$ 
20:  end if
21:   $\Sigma = \tilde{\Sigma}_{(1:N_s, 1:N_s)}$ 
22:  Compute orthogonality parameter,  $o$ , by (8.3.8)
23:  if  $|o| > \min(\epsilon_{SVD}, \epsilon_{ortho} \cdot N_h)$  then                                     ▷ Check orthogonality
24:    weightedGramSchmidt(U, M)
25:  end if
26:  if  $\Sigma_{(N_s, N_s)} < \epsilon_{SVD}$  then
27:     $\Sigma = \Sigma_{(1:N_s-1, 1:N_s-1)}$ ,  $U = U_{(:, 1:N_s-1)}$ 
28:     $N_s = N_s - 1$ 
29:  end if
30: end if
    
```


Moreover, let $\mathbf{e}_{\alpha_m} \in \mathbb{R}^{N_h}$ be the characteristic vector of point α_m , $P_m \in \mathbb{M}^{N_h \times m}$ be the matrix such that

$$P_m := [\mathbf{e}_{\alpha_1}, \mathbf{e}_{\alpha_2}, \dots, \mathbf{e}_{\alpha_m}], \quad (8.3.13)$$

\mathbf{c}_l be the solution of the linear system

$$(P_{l-1}^T L_{l-1}) \mathbf{c}_l = P_{l-1}^T \boldsymbol{\varphi}_l, \quad (8.3.14)$$

and

$$\mathbf{r}_l = \boldsymbol{\varphi}_l - L \mathbf{c}_l. \quad (8.3.15)$$

The procedure constructs a set of indices inductively on the input basis $\{\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_l\}$. The ordering of the input basis according to the dominant singular values is important and hence the POD basis is a suitable choice for this algorithm. The process starts from selecting the first interpolation index α_1 corresponding to the entry of the first input basis $\boldsymbol{\varphi}_1$ with largest magnitude. We select the remaining interpolation indices, α_l , for $l = 2, \dots, N_{MP}$, such that each of them corresponds to the entry with largest magnitude of the residual (8.3.15). The term \mathbf{r}_l can be viewed as the residual or the error between the input basis $\boldsymbol{\varphi}_l$ and its approximation $L \mathbf{c}_l$ from interpolating the basis $\{\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_{l-1}\}$ at the indices $\alpha_1, \dots, \alpha_{l-1}$. The linear independence of the input basis can be used to show that the DEIM procedure is indeed well-defined, i.e. $P_l^T L_l$ is nonsingular for all iterations. Moreover, the selected magic points are hierarchical and non-repeating.

Using this methodology, we select the cardinality of \mathcal{A} , N_{MP} , only. Then, Algorithm 9 automatically selects the magic points for the error estimator.

Algorithm 9 Error estimation points selection

Input POD modes, $\{\boldsymbol{\varphi}_l\}_{l=1}^{N_r}$; number of magic points, $N_{MP} \leq N_r$

- 1: $\alpha_1 = \arg \max_{i \in \mathcal{T}} |\boldsymbol{\varphi}_1|$
- 2: $l = 2$
- 3: **while** $l < N_{MP}$ **do**
- 4: Assemble L_{l-1} by (8.3.12)
- 5: Assemble P_{l-1} by (8.3.13)
- 6: Compute \mathbf{c}_l by solving (8.3.14)
- 7: Compute the residual (8.3.15)
- 8: Select the new point $\alpha_l = \arg \max_{i \in \mathcal{T}} |\mathbf{r}_i|$.
- 9: $l = l + 1$
- 10: **end while**

Notice that the computational cost of computing the error estimator (8.3.11) depends on the number of magic points, N_{MP} , that we select. However, this number is bounded by the number of RB, N_r . Then, since N_r is generally in the order of few dozens, there is no risk to exceed with the number of magic points.

8.3.3 Incremental Algorithm for the Computation of the Reduced Basis

Until now, in the present section, we derived a methodology for the incremental computation of the POD modes and an error estimator for the projection of a new IC, T_0^k , onto the POD basis. Now, we are ready to implement them in an algorithm to perform a new offline phase for the inverse solvers that we studied in Chapter 7.

In this new offline phase, we assume to have available a dataset of thermocouples measurements, $\hat{T}_{train}(\Psi, \Upsilon_{train})$, with $\Upsilon_{train} := \{\tau^0, \tau^1, \dots, \tau^{P_{train}}\}$, that comes from a previous run of the CC mold. Moreover, we select a projection tolerance, tol_{proj} , the number of RB, N_r , and the number of magic points, N_{MP} .

The novel algorithm starts by performing all the offline computations required to set up the FOM inverse solver. Later, we iterate over the measurements training set, $\hat{T}_{train}(\Psi, \Upsilon_{train})$, to perform a tranche of solutions of the inverse solver.

At the beginning of this training phase, we have an empty RB space. Then, we perform a first FOM iteration to generate snapshots and compute the first POD modes. In the following iterations, we start by computing the projection error, ϵ_{proj} , by (8.3.11). If ϵ_{proj} is above the user-defined tolerance, tol_{proj} , the RB space is not rich enough to represent the incoming IC, T_0^k , and we perform a new FOM iteration improving the POD space by Algorithm 8, computing the new magic points and assembling the ROM matrices (8.2.14). Otherwise, we use the ROM (8.2.15) and move to the next iteration.

We summarize this procedure in Algorithm 10 creating a single offline algorithm for all the inverse solvers of Chapter 7. As mentioned, this data-driven incremental procedure can be performed during a single offline phase that is computationally expensive and involves several full order computations. However, it is done once and for all before the machinery starts to work. In the next section, we use it to propose a novel inverse solver that exploits the ROM constructed in Algorithm 10 to speed up the heat flux estimation.

Algorithm 10 Offline phase for the creation of the RB space.

Input RBF shape parameter, η ; thermocouples measurement points, Ψ ; projection tolerance, tol_{proj} ; training set, $\hat{T}_{train}(\Psi, \Upsilon_{train})$; maximum number of POD basis, N_r ; IC, T_0 ;

- 1: Perform the offline phase of Algorithm 3, 4, 5 or 6
 - 2: Set $k = 1$
 - 3: **while** $k \leq P_{train}$ **do** ▷ ROM training
 - 4: **if** $N_s > 0$ **then**
 - 5: Compute projection error, ϵ_{proj} by (8.3.11)
 - 6: **end if**
 - 7: **if** $\epsilon_{proj} > \text{tol}_{proj}$ **or** $N_s = 0$ **then** ▷ RB space enrichment
 - 8: Compute $\mathbf{T}_{IC}^k[\mathbf{w}^k]$ by solving the FOM of Problem 7.6
 - 9: Use the computed solutions to enrich the POD space by Algorithm 8
 - 10: Compute the magic points using Algorithm 9
 - 11: Compute M_r , A_r and $\tilde{\mathbf{b}}_r$ by (8.2.14)
 - 12: **else** ▷ Use reduced-order model
 - 13: Compute $\tilde{\mathbf{T}}_{IC}^k$ by solving the ROM of Problem 7.6, (8.2.15)
 - 14: Assemble \mathbf{T}_{IC}^k by (8.2.10)
 - 15: **end if**
 - 16: **if** piecewise constant heat flux **then**
 - 17: Compute $\hat{\mathbf{w}}^k$ by solving (7.3.17)
 - 18: Compute $g_i^k(t)$ by (7.2.2)
 - 19: Compute the heat flux $\mathbf{g}(\mathbf{x}, t)$ for $t \in (\tau^{k-1}, \tau^k]$ by (7.2.1)
 - 20: Use (7.2.21) to compute $T^k[g^k]$
 - 21: **else if** piecewise linear heat flux **then**
 - 22: Compute $\hat{\mathbf{w}}^k$ by solving (7.3.19)
 - 23: Compute $g_i^k(t)$ by (7.2.4)
 - 24: Compute the heat flux $\mathbf{g}(\mathbf{x}, t)$ for $t \in (\tau^{k-1}, \tau^k]$ by (7.2.1)
 - 25: Use (7.2.39) to compute $T^k[g^k]$
 - 26: **end if**
 - 27: $k = k + 1$
 - 28: **end while**
-

8.4 REDUCED ORDER INVERSE SOLVERS

In the previous sections, we developed a technique for the data-based, automated construction of a POD-Galerkin ROM of Problem 7.6. First, we described the generation of a general ROM by standard POD-Galerkin. Then, due to the impossibility

of properly sampling a priori the parameter space, we proposed a methodology for the incremental construction of the RB space exploiting the sequentiality of our inverse problem. To make this technique more powerful and automated, we equipped it with an error estimator that can automatically and quickly determine whether the RB space is rich enough for the present iteration of the inverse solver.

Thanks to the developed methodologies, we are now ready to assemble a new version of Algorithms 3 and 4 that exploits MOR to speed up the computations in the online phases. The novel inverse solver starts by running the previously developed offline phase that we summarized in Algorithm 10. Later, in each iteration of the online phase, we read a new set of measurements coming from the thermocouples. Then, we compute the projection error estimator, ϵ_{proj} , by (8.3.11). If the training set was rich enough, the value of this error estimator will be below the defined tolerance and we can use the ROM of Problem 7.6. Otherwise, if the error estimator is above the tolerance, or the training dataset was not chosen accurately or an unusual event is happening. In this case, the algorithm requires the computation of a FOM solution for Problem 7.6 to further enrich the RB space. Once we have a solution to Problem 7.6, we proceed as in the full order inverse solvers. We summarize this novel inverse solver in Algorithms 11.

Algorithm 11 Reduced order inverse solver for the solution of Problem 7.2.

OFFLINE

Input RBF shape parameter, η ; thermocouples measurement points, Ψ, Υ ; projection tolerance, tol_{proj} ; training set, $\hat{T}_{train}(\Psi, \Upsilon_{train})$; maximum number of POD basis, N_r ; IC, T_0 ;

1: Run Algorithm 10

ONLINE

Input IC, T_0

1: Set $k = 1$
2: **while** $k \leq P_t$ **do**
3: Read the thermocouples measurements, $\hat{\mathbf{T}}^k$
4: Compute projection error, ϵ_{proj} by (8.3.11)
5: **if** $\epsilon_{proj} > \text{tol}_{proj}$ **then** ▷ Perform full-order solution
6: Compute $\mathbf{T}_{IC}^k[\mathbf{w}^k]$ by solving the FOM of Problem 7.6
7: Use the computed solutions to enrich the POD space by Algorithm 8
8: Compute the magic points using Algorithm 9
9: Compute M_r, A_r and $\tilde{\mathbf{b}}_r$ by (8.2.14)
10: **else** ▷ Use reduced-order model
11: Compute $\tilde{\mathbf{T}}_{IC}^k$ by solving (8.2.15)
12: Assemble \mathbf{T}_{IC}^k by (8.2.10)
13: **end if**
14: **if** piecewise constant heat flux **then**
15: Compute $\hat{\mathbf{w}}^k$ by solving (7.3.17)
16: Compute $g_i^k(t)$ by (7.2.2)
17: Compute the heat flux $\mathbf{g}(\mathbf{x}, t)$ for $t \in (\tau^{k-1}, \tau^k]$ by (7.2.1)
18: Use (7.2.21) to compute $T^k[g^k]$
19: **else if** piecewise linear heat flux **then**
20: Compute $\hat{\mathbf{w}}^k$ by solving (7.3.19)
21: Compute $g_i^k(t)$ by (7.2.4)
22: Compute the heat flux $\mathbf{g}(\mathbf{x}, t)$ for $t \in (\tau^{k-1}, \tau^k]$ by (7.2.1)
23: Use (7.2.39) to compute $T^k[g^k]$
24: **end if**
25: $k = k + 1$
26: **end while**

9 BENCHMARKS

To test the previously developed methodologies, we design different benchmark cases. Through these tests, we validate and analyze the performances of the direct and inverse solvers that we proposed in the previous chapters.

First, we consider in Section 9.1 the numerical solution of the direct Problem 6.1 on an academic benchmark case. Then, we design two benchmarks to perform different tests for the inverse problem solvers proposed in Chapters 7 and 8.

9.1 BENCHMARK 1

The benchmark presented in this section is an academic case. It is a unsteady-state heat conduction problem in a homogeneous isotropic solid occupying a rectangular parallelepiped domain. By carefully selecting the BCs on the faces of the parallelepiped and the IC, we are able to compute the analytical solution of the heat conduction problem. Our objective is to use this academic test to validate the numerical solution of the direct problem.

We consider the domain Ω_s of Figure 9.1.1. To have an analytical solution T_{an} in $\Omega_s \times (0, t_f]$, we consider the following modification of Problem 6.1.

Problem 9.1. Find T_s such that

$$\rho_s C_{p_s} \frac{\partial T_s}{\partial t} - k_s \Delta T_s = 0, \quad \text{in } \Omega_s \times (0, t_f], \quad (9.1.1)$$

with BCs and IC

$$\left\{ \begin{array}{ll} -k_s \nabla T_s \cdot \mathbf{n} = g_{an} & \text{on } \Gamma_{sin} \times (0, t_f], \quad (9.1.2) \\ -k_s \nabla T_s \cdot \mathbf{n} = q_L & \text{on } \Gamma_L \times (0, t_f], \quad (9.1.3) \\ & L \in \{I, II, III, IV\}, \quad (9.1.4) \\ -k_s \nabla T_s \cdot \mathbf{n} = h(T_s - T_f) & \text{on } \Gamma_{sf} \times (0, t_f], \quad (9.1.5) \\ T_s(\cdot, 0) = T_0 & \text{in } \Omega_s. \quad (9.1.6) \end{array} \right.$$



Now, let a, b, c, d be real constants. To have an analytical solution in Ω_s , we

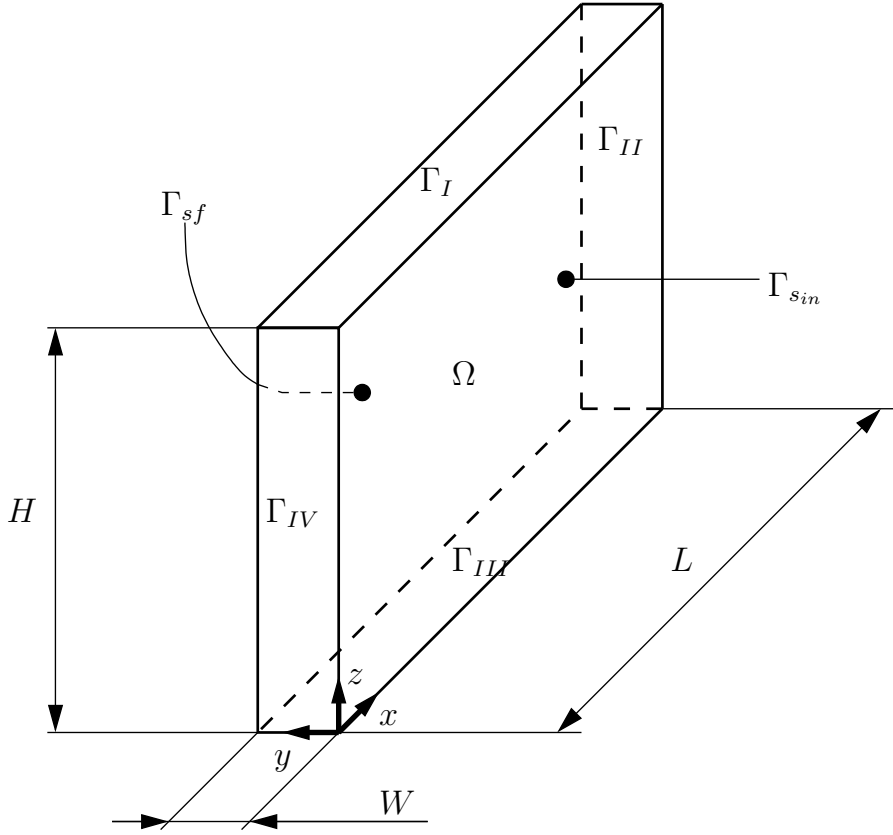


Figure 9.1.1: Schematic of the domain used for the unsteady benchmarks tests.

consider the following data as BCs for Problem 9.1,

$$\begin{aligned}
 q_I(\mathbf{x}, t) &= -k_s(bxe^{(t+y/\sqrt{v_s})} + 2cH), \\
 q_{III}(\mathbf{x}, t) &= k_s(bxe^{(t+y/\sqrt{v_s})} + 2cH), \\
 q_{II}(\mathbf{x}, t) &= -k_s(2aL + bze^{(t+y/\sqrt{v_s})}), \\
 q_{IV}(\mathbf{x}, t) &= k_s(2aL + bze^{(t+y/\sqrt{v_s})}),
 \end{aligned} \tag{9.1.7}$$

with

$$\begin{aligned}
 T_f(\mathbf{x}, t) &= \frac{k_s b}{h\sqrt{v_s}} xze^{(t+W/\sqrt{v_s})} + \\
 & USC \left(ax^2 + bxze^{(t+W/\sqrt{v_s})} + cz^2 + d + 2v_s(a+c)t \right), \\
 T_0(\mathbf{x}) &= ax^2 + bxze^{y/\sqrt{v_s}} + cz^2 + d, \\
 g_{an}(\mathbf{x}, t) &= k_s \left(\frac{b}{\sqrt{v_s}} xze^{(t+y/\sqrt{v_s})} \right),
 \end{aligned} \tag{9.1.8}$$

where $v_s := k_s/(\rho_s C_{p_s})$.

With these BCs, the solution to Problem 9.1 is

$$T_{an}(\mathbf{x}, t) = ax^2 + bxze^{(t+y/\sqrt{v_s})} + cz^2 + d + 2v_s(a + c)t. \quad (9.1.9)$$

We now perform numerical tests to compare the performances of different meshes and time discretization in solving Problem 9.1. Table 9.1 summarizes the parameters used for the computations.

Table 9.1: Parameters used to test of the numerical solution of the direct Problem 9.1.

Parameter	Value
Thermal conductivity, k_s	2.0 W/(m K)
Density, ρ_s	3.0 kg/m ³
Specific heat capacity, C_{p_s}	20.0 J/(kg K)
Heat transfer coefficient, h	7.0 W/(m ² K)
a	5 K/m ²
b	10 K/m ²
c	15 K/m ²
d	20 K/m ²
L	1 m
W	1 m
H	1 m
Final time, t_f	1 s

Notice that this is a very challenging test because we have an exponential solution both in time and in the y -direction. Moreover, the exponential dependence in space is scaled by $1/\sqrt{v_s}$. So, small values of v_s would lead to very strong gradients in the y -direction. However, high values of v_s would require very fine time discretization to avoid prohibitive Courant number values, where the Courant number is given by

$$Co = v_s \Delta t \left(\frac{1}{\Delta x} + \frac{1}{\Delta y} + \frac{1}{\Delta z} \right). \quad (9.1.10)$$

We test the accuracy of the numerical schemes for the direct Problem 9.1 refining both the time and space discretization. Figure 9.1.2 (a) summarizes the relative errors, between computed and analytical temperature, obtained refining the mesh, and keeping the same number of elements on each parallelepiped side. Figure 9.1.2 (b) shows the performances of the time discretization scheme for different time-step sizes.

By looking at the results, we appreciate that, for high values of the Courant number, the gain of increasing space discretization is reduced, and the error remains al-

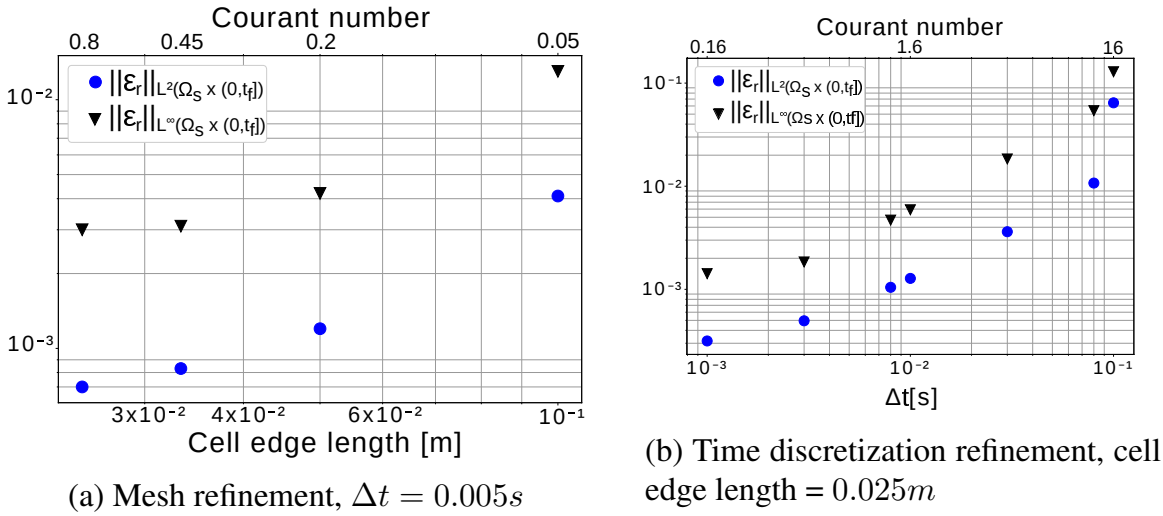


Figure 9.1.2: Behavior of the L^2 - and L^∞ -norm relative errors between the computed and analytical solution of Problem 9.1, with the refinement of space and time discretization.

most constant. Similarly, coarsening the time discretization causing high values of the Co leads to a steep increase of the error.

9.2 BENCHMARK 2

In this section, we test the performances of the inverse solvers proposed in Chapter 7 in the reconstruction of a linear in time heat flux, which is non-linear in space.

9.2.1 Setup of the Test Case

To design a numerical test case for the inverse problem, we proceed as follows: we arbitrarily define a boundary heat flux, $g_t(\mathbf{x}, t)$, and the thermocouples positions, Ψ , and sampling frequency, f_{samp} . Then, we solve the direct Problem 6.1 associated with $g_t(\mathbf{x}, t)$ in the time domain $(0, t_f]$, obtaining the related temperature field. Finally, we use its values at the thermocouples points and sampling times as input measurements to the inverse problem, $\hat{\mathbf{T}}$. Using this approach, we are able to analyze the inverse problem performance in the reconstruction of the boundary heat flux, $g_t(\mathbf{x}, t)$.

Table 9.2 shows the geometrical and physical parameters selected for the present benchmark case. In the attempt to mimic the real industrial situation of estimating the boundary heat flux in a plate of a CC mold, these parameters and the thermocouple locations (see Figure 9.2.1) are close to real industrial values. Similarly, we use the

Table 9.2: Geometrical and physical parameters used for the inverse unsteady test cases, Benchmark 2 and 3.

Parameter	Value
Thermal conductivity, k_s	383 W/(mK)
Density, ρ_s	8940 kg/m ³
Specific heat capacity, C_{p_s}	390 J/(kg K)
Heat transfer coefficient, h	5.66e4 W/(m ² K)
Water temperature, T_f	350 K
Initial condition, T_0	350 K
L	2 m
W	0.1 m
H	1.2 m
Sampling frequency, f_{samp}	1 Hz
a	1100
b	1200
c	3000
Final time, t_f	50 s

computational domain in Figure 9.1.1 where L , W and H are set as in a real mold plate.

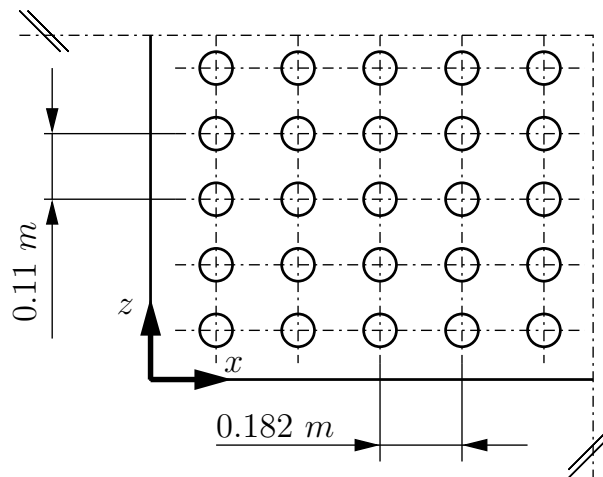


Figure 9.2.1: Position of the 100 thermocouples at the plane $y = 0.02$ m for the inverse solver tests.

To test the effect of the regularization by discretization, we use different space and time discretizations. For the time discretization, we use homogeneous time discretization with $\Delta t = 0.1s, 0.2s, 0.25s$ and $0.5s$. For the space discretization, we use the uniform, structured, orthogonal, hexahedral meshes presented in Table 9.3.

For this test case, we select the heat flux g_t to be linear in time and quadratic in

Table 9.3: Summary of the different meshes used in the unsteady Benchmark 2.

	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5
Number of elements	1.7e5	4.5e4	2.1e4	7.5e3	1.5e3

space as shown in Table 9.4. Moreover, to analyze the performance of the inverse solvers, we introduce the relative error

$$e_{rel}(\mathbf{x}, t) := \frac{g_t(\mathbf{x}, t) - g(\mathbf{x}, t)}{g_t(\mathbf{x}, t)}. \quad (9.2.1)$$

Table 9.4: Details of the true heat flux used in the unsteady Benchmark 2.

Heat flux	Value
$g(\mathbf{x})$	$bz^2 + c$
$g_t(\mathbf{x}, t)$	$-k_s (0.5tg(\mathbf{x}) + g(\mathbf{x})) \text{ W/m}^2$

9.2.2 Effect of Time and Space Discretization Refinement

Now, provided all the details for the benchmark setup, we can proceed presenting the results. At first, we show the effects of mesh and time discretization refinement. To do it, we do not add any noise to the temperature measurements and do not apply any regularization in the solution of the linear systems solving them by a LU factorization with full pivoting.

We start by analyzing the case in which we minimize the functional S_1^k (i.e. $p_g = 0$). First, we show in Figure 9.2.2 the maximum and mean value of the L^2 - and L^∞ -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, as the time and space discretization changes for Algorithm 3 (i.e. piecewise constant approximation in time of the heat flux).

From the figures, we appreciate on one side that the time discretization coarsening has very little effects on Algorithm 3 with as small decrease of the error as Δt increases. On the other, the space discretization does not have any effect on this inverse solver.

We now perform the same test for the piecewise linear time approximation of Algorithm 4. Similarly, Figure 9.2.3 shows the maximum and mean value of the L^2 - and L^∞ -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, as the time and space discretization changes.

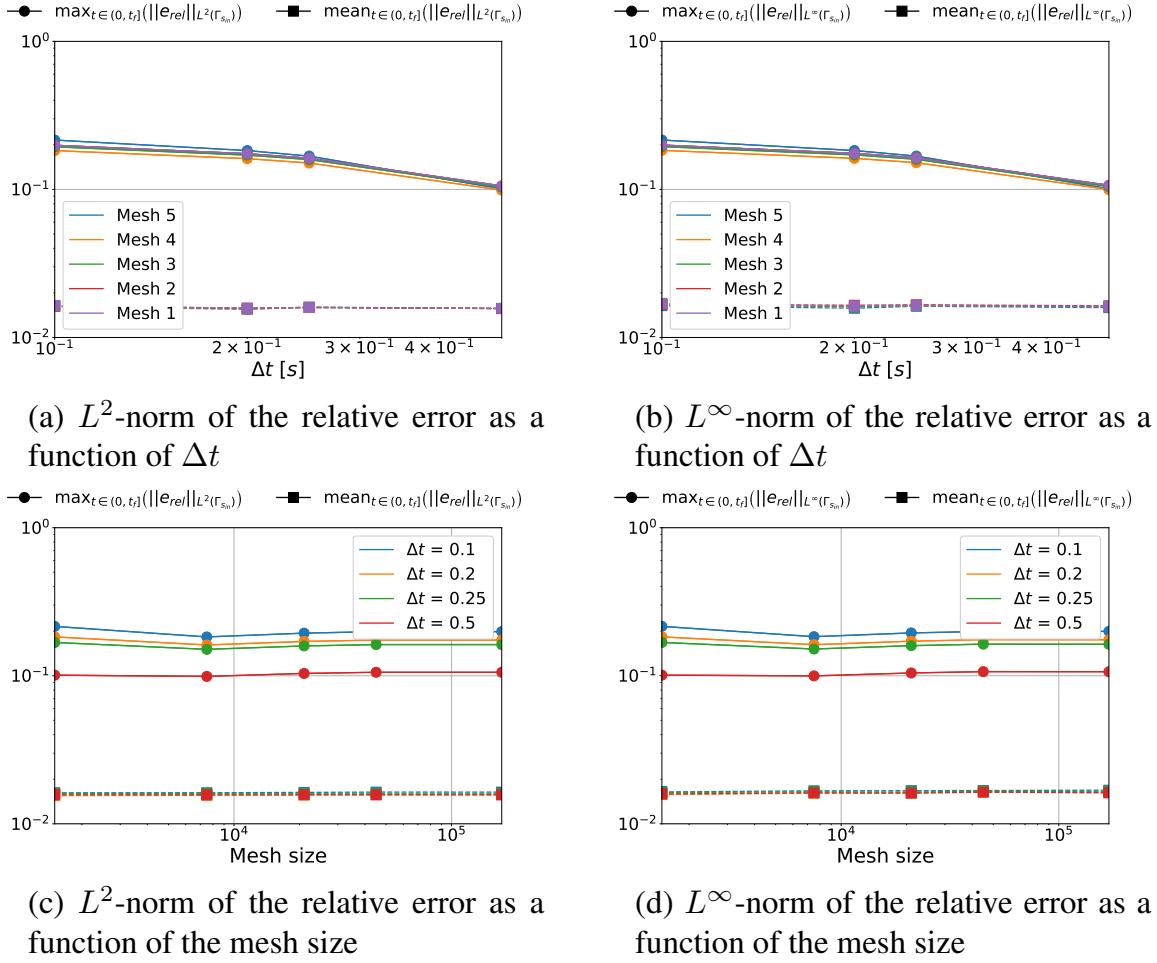
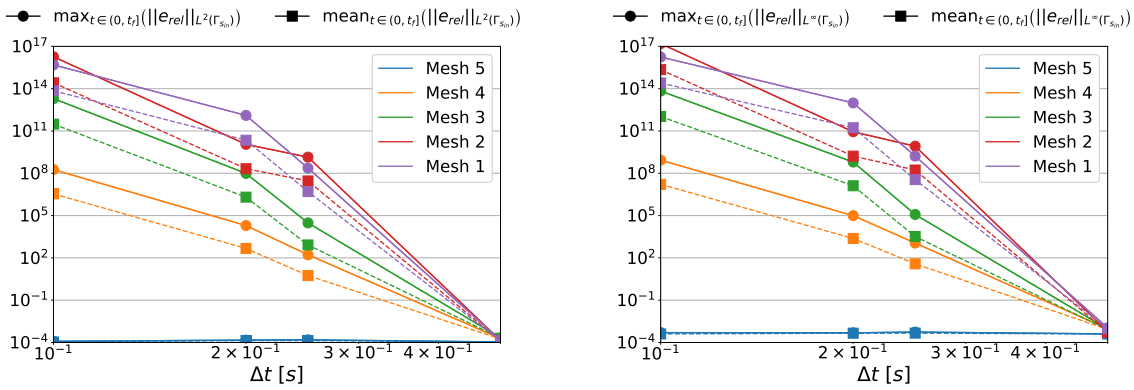


Figure 9.2.2: Maximum and mean values of the L^2 - and L^∞ -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, for Benchmark 2 as the time and space discretization changes for Algorithm 3 (piecewise constant time approximation of the heat flux and $p_g = 0 \frac{K^2}{W^2}$).

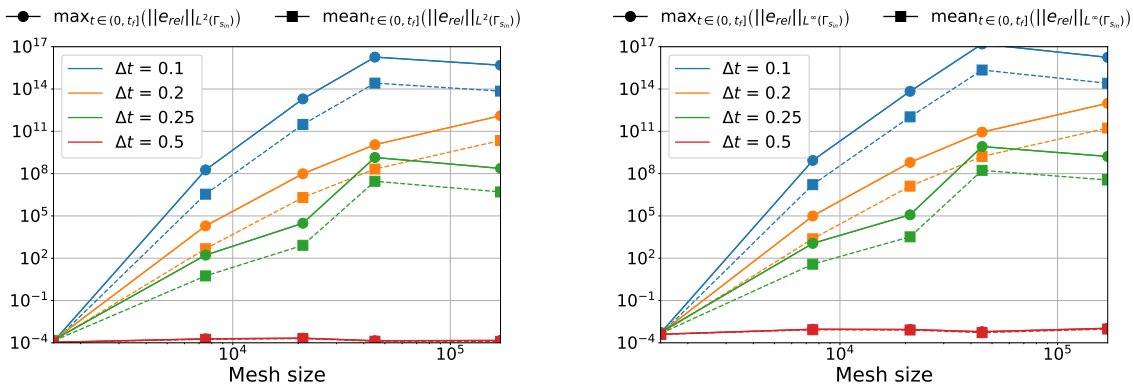
In this setting, the obtained results are very different from the previous case. First of all, we notice a massive influence of both the space and time discretization refinement on the performances of the inverse solver. As anticipated in Section 7.4, the regularization by discretization plays an important role as the algorithm performances are improved by several orders of magnitude by the coarsening of the discretization. Moreover, when comparing the results for Algorithm 3 and 4, we notice that the piecewise linear solver is able to outperform the constant one by three orders of magnitude but is also very unstable depending on the discretization.

To better understand the behaviour of this inverse solver, Figure 9.2.4 illustrates the L^2 -norm of the relative error, e_{rel} , as a function of time for mesh 3 with different Δt . From these results, we see that the high errors shown in Figure 9.2.3 are caused



(a) L^2 -norm of the relative error as a function of Δt

(b) L^∞ -norm of the relative error as a function of Δt



(c) L^2 -norm of the relative error as a function of the mesh size

(d) L^∞ -norm of the relative error as a function of the mesh size

Figure 9.2.3: Maximum (circles) and mean (squares) values of the L^2 - and L^∞ -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, for Benchmark 2 as the time and space discretization changes for Algorithm 4 (piecewise linear time approximation of the heat flux and $p_g = 0 \frac{K^2}{W^2}$).

by diverging oscillations in the algorithm. However, we also notice from the figure that, coarsening the time discretization, monotonically reduces such instability until achieving a stable solution, eventually.

9.2.3 Effect of Cost Functional Parameter, p_g

In this section, we analyze the role that the cost functional parameter, p_g , in (7.1.8), has on the performance of the proposed inverse solvers. To do it, we solve several times this benchmark case using the different meshes of Table 9.3 and dif-

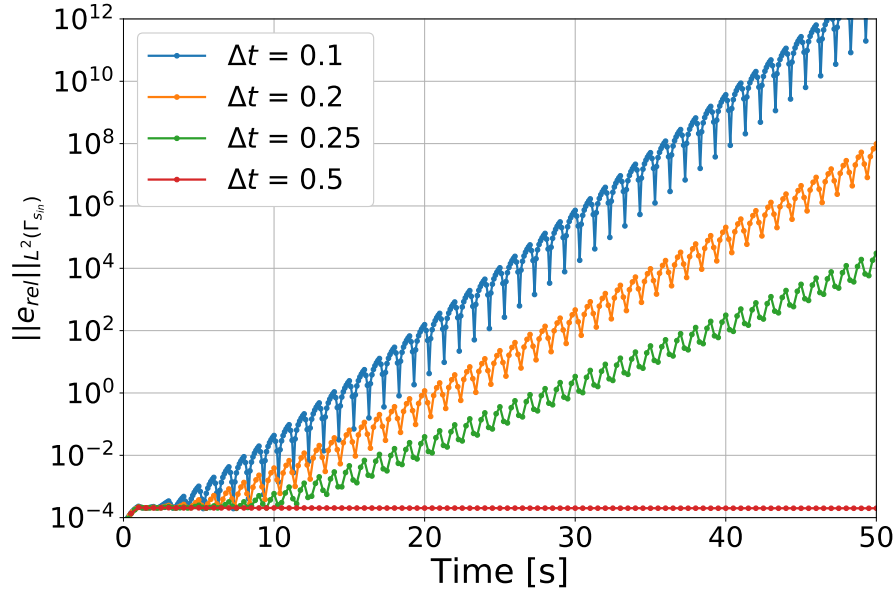


Figure 9.2.4: L^2 -norm of the relative error, e_{rel} , in Benchmark 2 for Algorithm 4 (piecewise linear time approximation of the heat flux and $p_g = 0 \frac{K^2}{W^2}$). The presented results are obtained with Mesh 3.

ferent timestep sizes. Then, we plot the maximum and mean value of the L^2 -norm of the relative error, e_{rel} , over the entire interval $t = (0, t_f]$ as a function of the cost functional parameter, p_g .

We start with Algorithm 5 (i.e. piecewise constant approximation in time of the heat flux). Figure 9.2.5 shows the obtained results for different timestep sizes and a fixed space discretization.

From the results, we notice that increasing the value of p_g monotonically decreases the quality of the reconstruction. Moreover, it is true for all considered Δt with a slight improvement of the performances as the time discretization gets coarser.

Now, we perform a similar test but this time we keep $\Delta t = 0.25$ s and test the different meshes of Table 9.3. We illustrate in Figure 9.2.6 the obtained results.

This figure confirms that Algorithm 5 is badly affected by the implementation of the second term of (7.1.8). In fact, its performance dramatically deteriorates as soon as this term begins to play a role (i.e. $p_g \gtrsim 10^{-12} K^2/W^2$). Moreover, the results are almost independent from the discretization refinement. This further confirms the insensibility of this algorithm from the used discretization.

We continue by performing the same kind of tests on Algorithm 5 (i.e. piecewise linear approximation of the heat flux in time). We start by testing different timestep

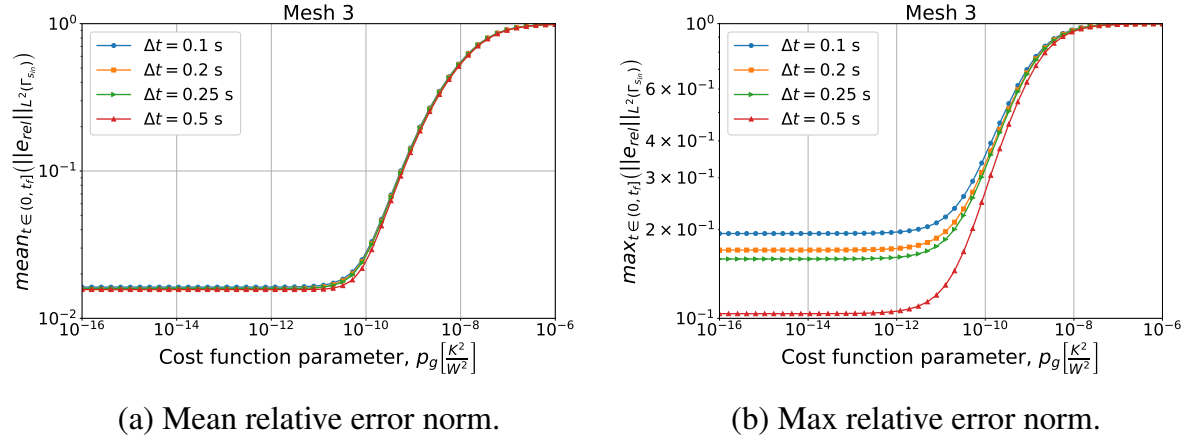


Figure 9.2.5: Mean and maximum values of the L^2 -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, for Benchmark 2 as the value of the cost function parameter, p_g , changes for Algorithm 5 (piecewise constant time approximation of the heat flux). We show the results for Mesh 3 and different Δt .

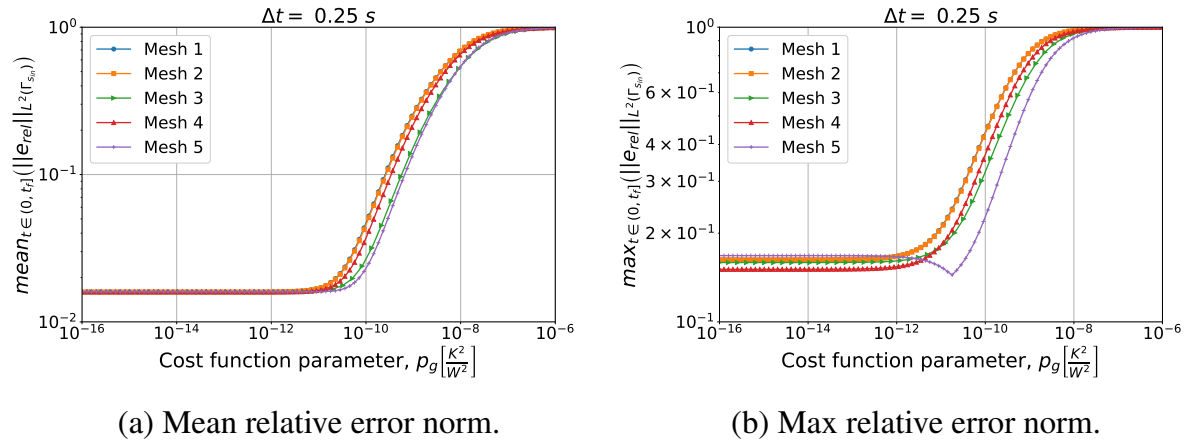


Figure 9.2.6: Mean and maximum values of the L^2 -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, for Benchmark 2 as the value of the cost function parameter, p_g , changes for Algorithm 5 (piecewise constant time approximation of the heat flux). We show the results for $\Delta t = 0.25$ s and different meshes.

sizes while using Mesh 3 for the space discretization. We present the results in Figure 9.2.7.

At first, we notice that this algorithm has a very different behaviour with respect to the piecewise constant case. In this case, the timestep size dramatically affects the results. We can depict two different behaviors as p_g changes for a chosen Δt . In the first one (i.e. $\Delta t = 0.1$ s, 0.2 s and 0.25 s), the inverse solver is very unstable and provides completely useless solutions for low values of p_g (i.e. $p_g \lesssim 10^{-12} K^2/W^2$).

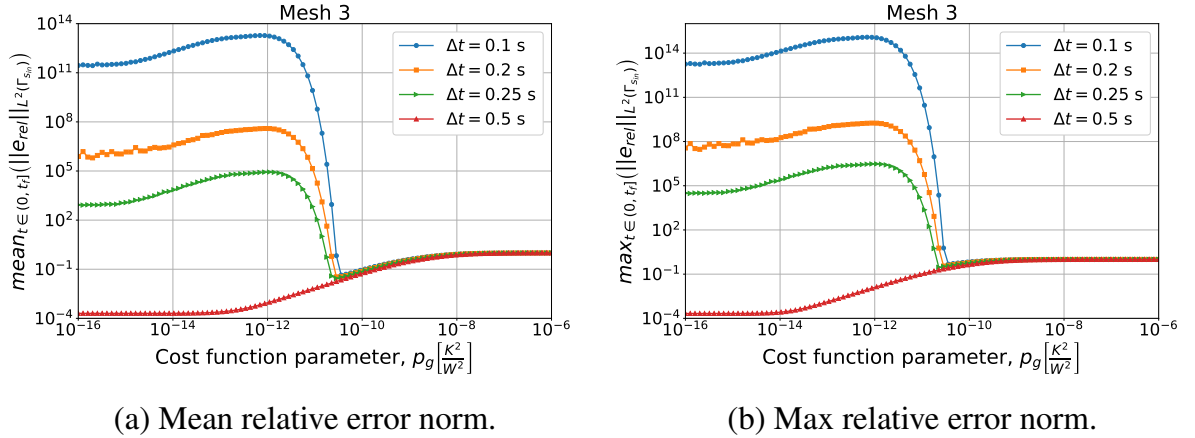


Figure 9.2.7: Mean and maximum values of the L^2 -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, for Benchmark 2 as the value of the cost function parameter, p_g , changes for Algorithm 6 (piecewise linear time approximation of the heat flux). We show the results for Mesh 3 and different Δt .

As p_g increases, the quality of the approximation rapidly rises up until the error reaches a minimum. Here, we have stable solutions and a good approximation of the heat flux. For higher values of p_g , the error monotonically increases until it reaches a plateau at 100%.

On the other hand, we have a different behaviour for $\Delta t = 0.5$ s. In this case, the inverse solver performs similarly to the piecewise constant case, but the quality of the estimation is by almost two orders of magnitude better. Then, we have stable and accurate solutions for low values of p_g . For $p_g \gtrsim 10^{-12} K^2/W^2$ we have a monotonic degradation of the heat flux estimation until we reach the 100% plateau.

It is interesting to notice that the second term in the functional S_2^k can make the solver insensible to the discretization refinement. In fact, after a certain value of p_g the relative error norm for the different Δt are almost coincident.

We can see a similar behaviour in Figure 9.2.8 where we show the results obtained refining the mesh and keeping $\Delta t = 0.25$ s. Also in this case, we notice the two previously described behaviors with the coarsest mesh being always stable and providing the best results for the lowest values of p_g .

These statements are remarked by the results shown in Figure 9.2.9, where we show the results of the same test as in Figure 9.2.3 but for $p_g = 5e - 11 \frac{K^2}{W^2}$. The obtained results confirm that for some values of p_g we can obtain a stable solver with a moderate dependency on the discretization refinement.

To conclude this analysis, we test the discretization and p_g selection method of Algorithm 7 in this benchmark case. We use the virtual thermocouples measurements

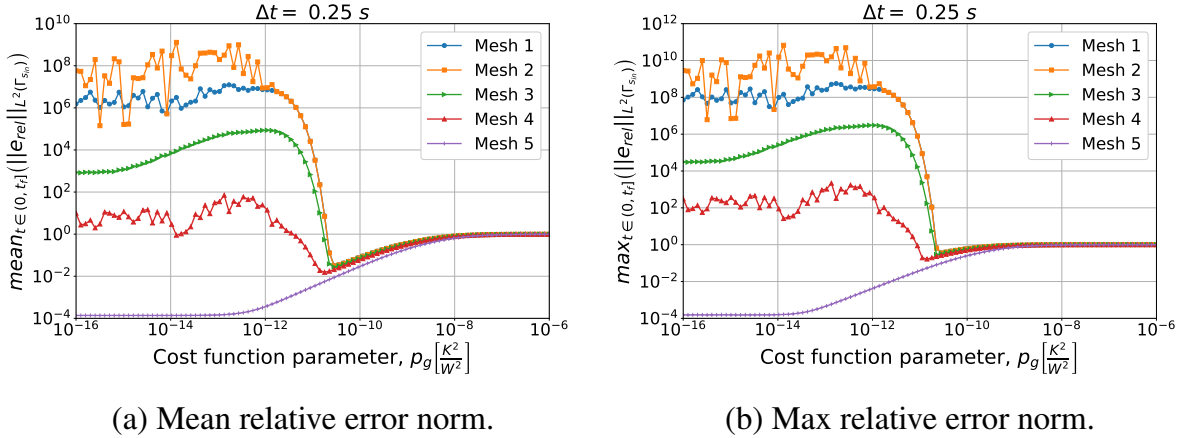


Figure 9.2.8: Mean and maximum values of the L^2 -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, for Benchmark 2 as the value of the cost function parameter, p_g , changes for Algorithm 6 (piecewise linear time approximation of the heat flux). We show the results for $\Delta t = 0.25s$ and different meshes.

as input training dataset for the algorithm, \hat{T}_{train} . In the test, the algorithm has to select a combination of mesh, timestep size and p_g that provides stable and accurate solutions to this inverse problem. Before presenting the results of Algorithm 7, we show in Figure 9.2.10 the mean value of the temperature discrepancy functional S_1^k , defined in (7.1.7), as function of p_g for different meshes and Δt .

These results show that m_S in (7.5.4) behaves similarly to the relative error (9.2.1) as function of p_g . Notice that it presents the same two behaviors that we previously described for the relative error depending on the used discretization setup. Moreover, the m_S minima are close to those of the relative error. For these reasons, in Section 7.5, we used this result in the creation of the selection criteria for the p_g as well as for the mesh and the Δt .

That said, we are now ready to test Algorithm 7. With respect to its implementation, in step 16 of the algorithm, we use the Nelder-Mead method to find the p_g that minimizes S_1^k (see [110]). To start the algorithm, we set $p_g^0 = 1e - 7 \frac{K^2}{W^2}$. Table 9.5 summarizes the algorithm behaviour.

Table 9.5: Test of Algorithm 7 in the unsteady Benchmark 2.

Iteration	Mesh	Δt [s]	$p_g \frac{K^2}{W^2}$	$mean_k(S_1^k) [K^2]$
0	5	0.5	$1e - 7$	$6.9e3$
1	5	0.5	$3.2e - 21$	$3.9e - 1$

From the results in the table, we appreciate that the algorithm chooses the coars-

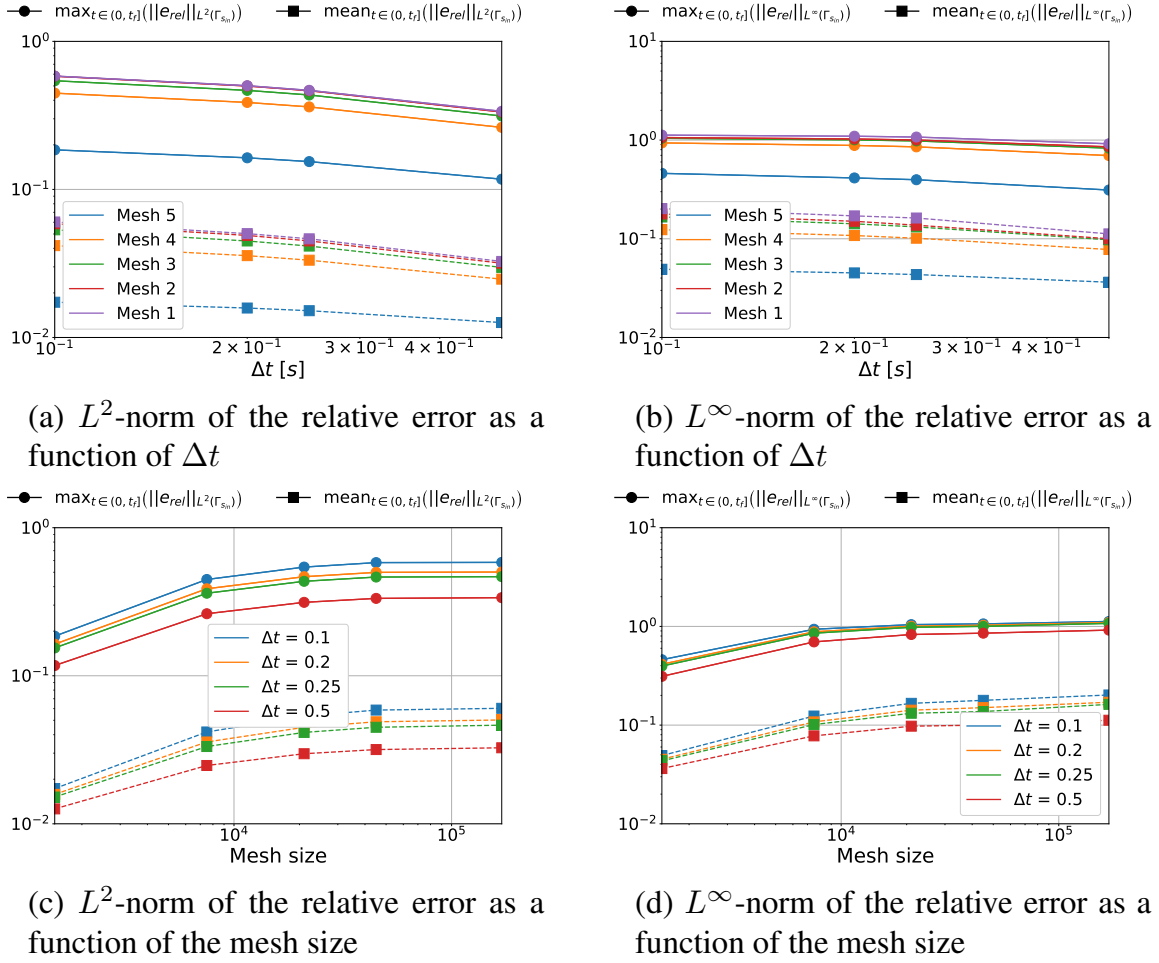


Figure 9.2.9: Maximum (circles) and mean (squares) values of the L^2 - and L^∞ -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, for Benchmark 2 as the time and space discretization changes for Algorithm 6 (piecewise linear time approximation of the heat flux and $p_g = 5e - 11 \frac{K^2}{W^2}$).

est discretization since the first iteration. Then, it looks for the p_g that minimizes S_1^k for this discretization and, not finding a better discretization setup for this value of p_g , exits the process. Comparing the obtained results to the relative error plots of Figures 9.2.7 and 9.2.8, we confirm that the algorithm is selecting the best configuration in between all the available.



9.3 BENCHMARK 3

In this section, we present a benchmark case which is designed to test the performances of the ROM inverse solver Algorithm 11 of Chapter 8.

For this reason, we require more complexity and time-space evolution in the heat

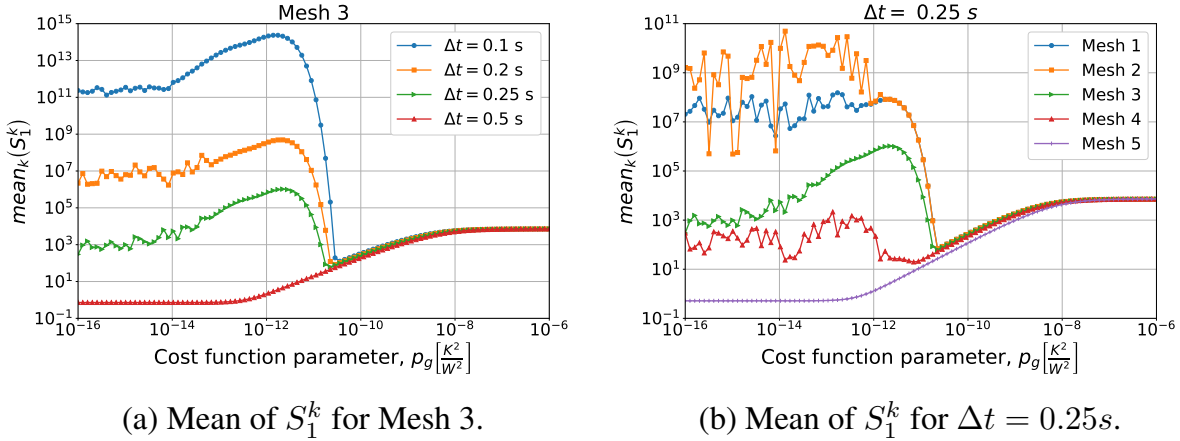


Figure 9.2.10: Mean values of S_1^k for $1 \leq k \leq P_t$, for Benchmark 2 as the value of the cost function parameter, p_g , changes. The results are obtained using Algorithm 6 (piecewise linear time approximation of the heat flux). We show the results for Mesh 3 and different Δt in (a), and for $\Delta t = 0.25$ and different meshes in (b).

flux with respect to the previous benchmark case. In particular, we use the same geometrical and physical parameters as in the previous Benchmark 2 but we choose a very non linear in time true heat flux, g_t , as in Table 9.6. In fact, by choosing such heat flux, we mean to design a test case that allows us to study the performances of the incremental POD algorithm and the related ROM in Algorithm 11. Then, a level of complexity in the space and time evolution of the heat flux is necessary to avoid the trivial case in which very few POD basis are sufficient to represent all the input parameters of Problem 7.6. Finally, we test in Section 9.3.2 the effect that adding noise to the thermocouples measurements has on the estimation of the heat flux.

Table 9.6: Parameters used for the unsteady Benchmark 3.

Parameter	Value
$g_1(\mathbf{x})$	$bz^2 + c$
$g_2(\mathbf{x})$	$\frac{10c}{1+(x-1)^2+z^2}$
Heat flux, $g_t(\mathbf{x}, t)$	$-k_s \left[g_1 + \frac{g_1}{2} \sin \left(2\pi f_{max} \frac{t^2}{t_f} \right) + g_2 e^{-0.1t} \right] W/m^2$
Maximum frequency, f_{max}	0.1 Hz



9.3.1 Comparison Between Full and Reduced Order Inverse Solvers

The reduced order Algorithm 11 requires some additional user defined input parameters with respect to the full order ones (i.e. Algorithms 3 and 4). Their selection

is problem dependent and they have to be tuned on the specific problem. For the present test case, we use the values summarized in Table 9.7.

Table 9.7: Input parameters for the reduced inverse solver, Algorithm 11 used in the solution of the unsteady Benchmark 3.

Parameter	Value
SVD tolerance, ϵ_{SVD}	$1e-8$
Orthogonality tolerance, ϵ_{ortho}	$1e-16$
Number of magic points, N_{MP}	18
$\tilde{\mathbf{T}}_{IC}$ projection tolerance, tol_{proj}	$1e-9$
Mesh	Mesh 3
Δt	0.5 s

Figure 9.3.1 and 9.3.2 show the obtained results for the piecewise constant and linear time approximation of the heat flux respectively. To understand the behaviour of the different solvers, we plot in Figure 9.3.1(a) and 9.3.2(a) the value of the L^2 - and L^∞ -norm of the relative error (9.2.1) in $(0, t_f]$. Moreover, we choose a point on Γ_{sin} , $\mathbf{x} = (1, 0.02, 0.6)$, and we plot in Figure 9.3.1(b) and 9.3.2(b) the value of the true and reconstructed heat fluxes at this point along time.

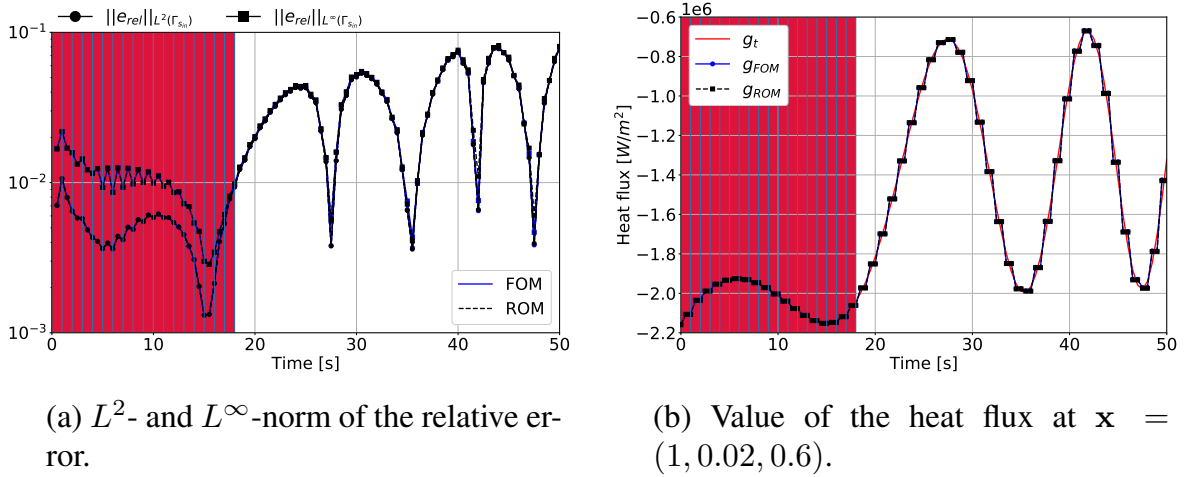
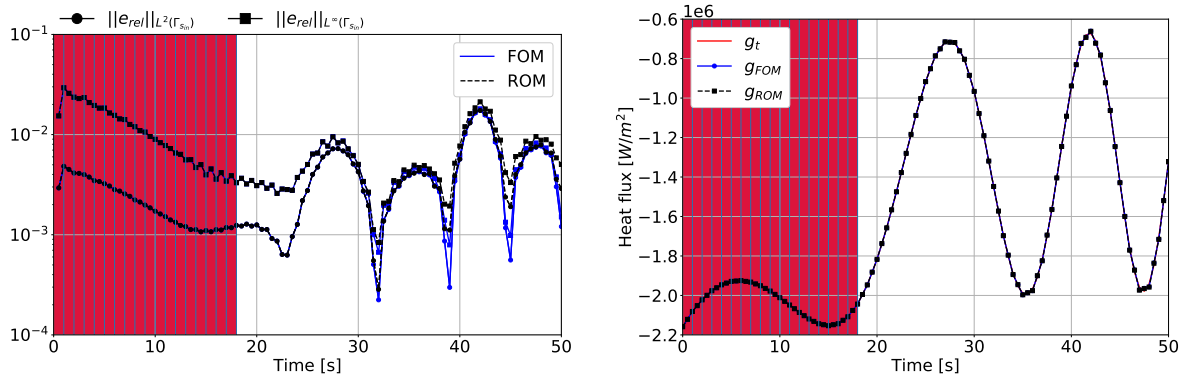


Figure 9.3.1: Comparison between FOM (Algorithm 3) and ROM (Algorithm 11) inverse solvers in the piecewise constant time approximation case for Benchmark 3. The red bars indicate the full order solution steps to enrich the POD space ($\epsilon_{proj} > tol_{proj}$).

If we look at Figure 9.3.1 and 9.3.2, we notice very similar results. First of all, the reduced order inverse solvers provides almost identical performances as the full order one. Moreover, we can appreciate that both for the piecewise constant and linear approximation, Algorithm 11 performs all the full order, RB space enriching, runs



(a) L^2 - and L^∞ -norm of the relative error. (b) Value of the heat flux at $\mathbf{x} = (1, 0.02, 0.6)$.

Figure 9.3.2: Comparison between FOM (Algorithm 4) and ROM (Algorithm 11) inverse solvers in the piecewise linear time approximation case for Benchmark 3. The red bars indicate the full order solution steps to enrich the POD space ($\epsilon_{proj} > \text{tol}_{proj}$).

at the beginning of the simulation (red rectangles in the figures) and then continues with the ROM. These are very promising results indicating that the incremental POD enrichment is properly working providing a reliable ROM for Problem 7.6.

To conclude this analysis, Table 9.8 illustrates the average CPU time required for the computation of one step of the inverse solver at full and reduced order in the solution of Benchmark 3. We can see from the table that the speed up is of almost 12 times for this test case. Moreover, while the full order CPU requirement could increase significantly in the true industrial setting overcoming the real-time threshold of 1 s, the ROM online phase is independent from the mesh size. Thus, it will provide real-time solutions also for more demanding meshes.

Table 9.8: Average CPU time required for the computation of one step of the inverse solver at full and reduced order in the solution of Benchmark 3 using Mesh 3 and the piecewise linear approximation in time of the heat flux. All computations are performed in serial on a Intel® Core™ i7-8550U CPU processor.

	FOM	ROM
Arg. CPU time	1613 ms	135 ms

9.3.2 Effects of Measurements Noise and Regularization

In this last numerical tests section, we test the effect that adding noise to the measurements vector, $\hat{\mathbf{T}}$, has in the performances of Algorithm 3 and 4. From the

industrial point of view, this analysis is of particular interest for our application since in the real case, thermocouples measurements are affected by noise.

We perform this analysis by adding to the measurements vector the Gaussian random noise $\boldsymbol{\eta} = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, where $\boldsymbol{\mu} \in \mathbb{R}^M$ is the mean vector and $\Sigma \in \mathbb{M}^{M \times M}$ is the covariance matrix. Then, we have

$$\hat{\mathbf{T}}_{\eta}^k = \hat{\mathbf{T}}^k + \boldsymbol{\eta}. \quad (9.3.1)$$

In particular, we choose $\boldsymbol{\eta}$ to be an IID random variable with zero mean, i.e. $\boldsymbol{\eta} = \mathcal{N}(\mathbf{0}, \omega^2 I)$, where ω denotes the noise standard deviation. To study the effect of noise, we perform several solutions of the inverse problem using $\hat{\mathbf{T}}_{\eta}^k$ as thermocouples' measurements. For each test, we compute 200 samples.

We show in Figures 9.3.3 and 9.3.4 the obtained results for the piecewise constant and linear algorithm, respectively. In particular, we illustrate for each of them the mean values over the samples of the mean and maximum of the relative error (9.2.1) in $(0, t_f]$ (with 90% quantile bars) for different values of the noise standard deviation, ω . The figure compares the results obtained using LU with full pivoting and TSVD with different values for the regularization parameter α_{TSVD} .

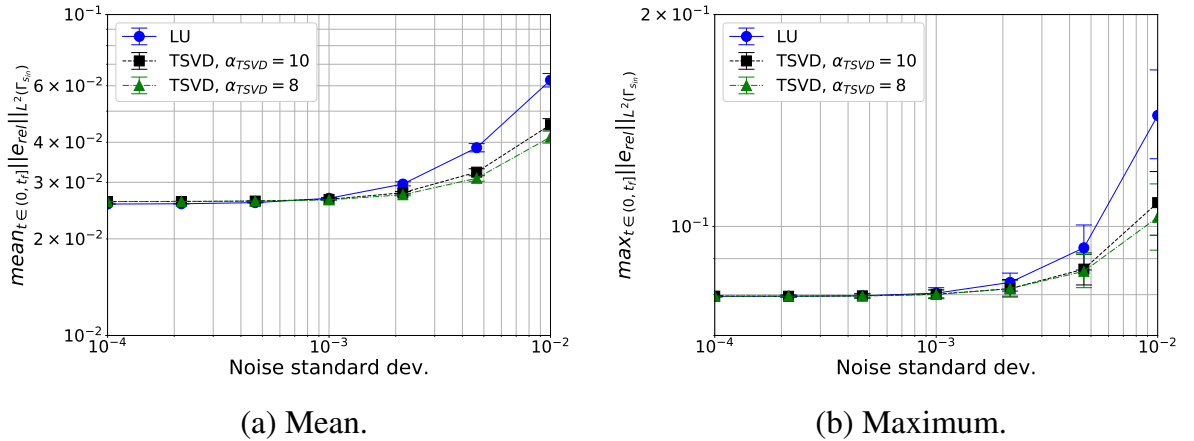


Figure 9.3.3: Effect of the noise in the temperature measurements for the piecewise constant case. In the figures, we show the mean (a) and maximum (b) values of the relative error (9.2.1) in $(0, t_f]$ for different values of the noise standard deviation and using both LU with full pivoting and TSVD for the solution of inverse problem linear system (7.2.33). For each case, we performed 200 runs. The markers show the mean values while the bars are the 90% quantiles. In these computations, we considered $p_g = 0 \frac{K^2}{W^2}$.

The results show a very different dependency from the measurement noise in the two algorithm. The piecewise constant Algorithm 3 shows in Figure 9.3.3 to be

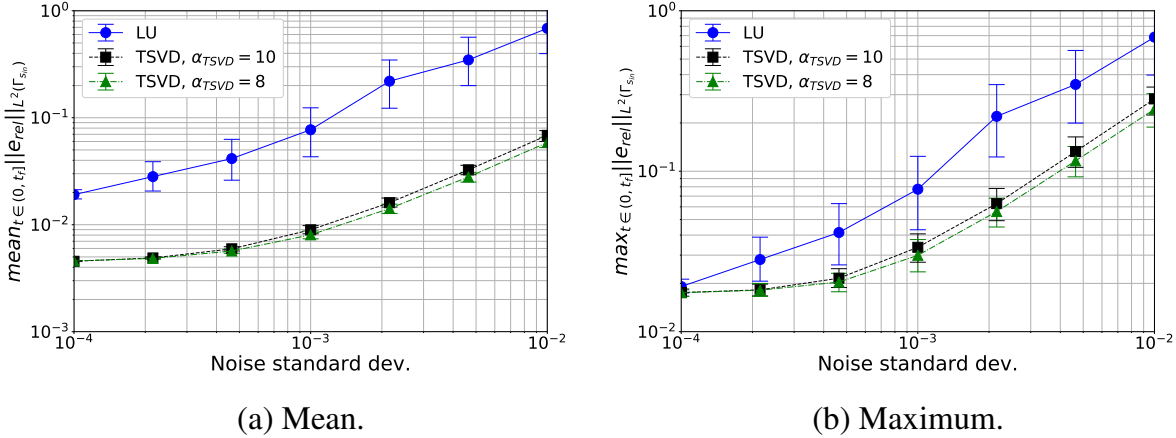


Figure 9.3.4: Effect of the noise in the temperature measurements for the piecewise linear case. In the figures, we show the mean (a) and maximum (b) values of the relative error (9.2.1) in $(0, t_f]$ for different values of the noise standard deviation and using both LU with full pivoting and TSVD for the solution of inverse problem linear system (7.2.51). For each case, we performed 200 runs. The markers show the mean values while the bars are the 90% quantiles. In these computations, we considered $p_g = 0 \frac{K^2}{W^2}$.

quite robust with respect to these levels of noise. Moreover, the TSVD is effective in reducing the noise propagation and we are able to keep a reasonable level of accuracy.

On the other hand, the piecewise linear Algorithm 4 is much more affected by the noise. By using the TSVD regularization, we have an improvement of the noise robustness. However, the error rate of increase is much higher than for the piecewise constant solver.

9.3.3 Effect of Cost Functional Parameter, p_g

Also for this benchmark case, we test the effects that the parameter p_g in (7.1.8) has on the inverse solvers performances. We do it by performing the same tests of Section 9.2.3 but for the present test case. In particular, we solve this inverse problem using all the meshes in Table 9.3 and $\Delta t = 0.1, 0.2, 0.25$ and 0.5 s, for $1e - 16 \leq p_g \leq 1e - 6 \frac{K^2}{W^2}$. We use both the piecewise constant and linear approximation of the heat flux in Algorithm 5 and 6, respectively. Notice that in these tests, we use the full order algorithms.

First, we present in Figures 9.3.5 and 9.3.6 the results obtained using the piecewise constant approximation algorithm. These plots show the mean and maximum values of the L^2 -norm of the relative error, e_{rel} , in the interval $(0, t_f]$ as functions of p_g changing the mesh and the Δt , respectively.

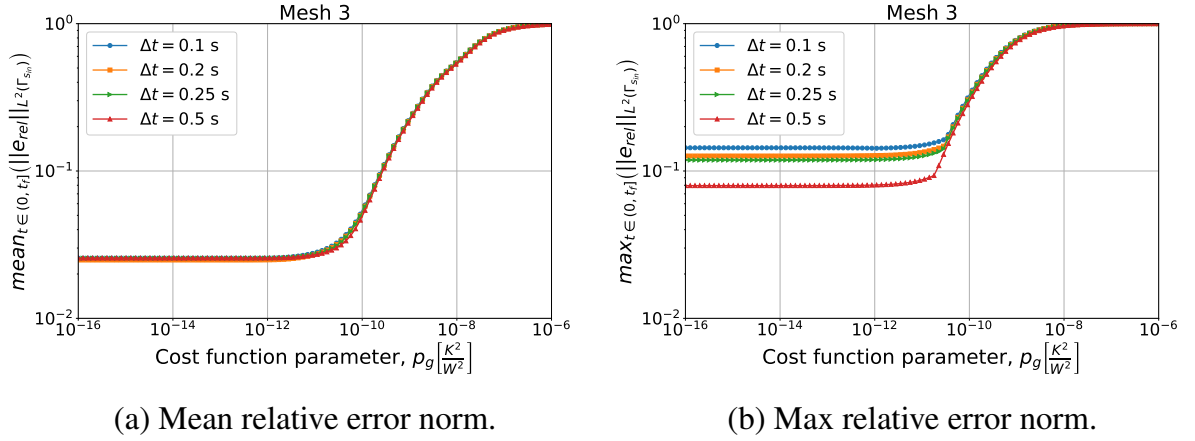


Figure 9.3.5: Mean and maximum values of the L^2 -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, for Benchmark 3 as the value of the cost function parameter, p_g , changes for Algorithm 5 (piecewise constant time approximation of the heat flux). We show the results for Mesh 3 and different Δt .

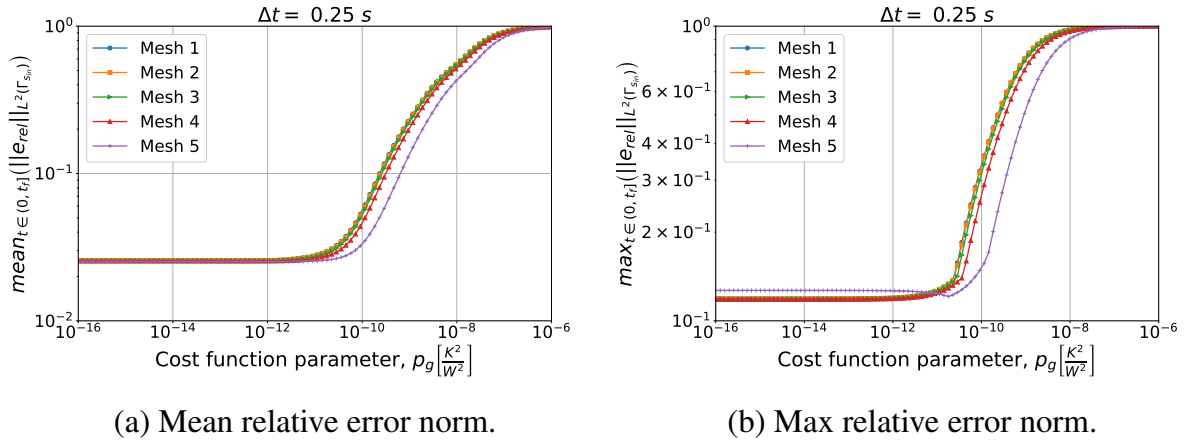


Figure 9.3.6: Mean and maximum values of the L^2 -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, for Benchmark 3 as the value of the cost function parameter, p_g , changes for Algorithm 5 (piecewise constant time approximation of the heat flux). We show the results for $\Delta t = 0.25s$ and different meshes.

From the presented results, we notice that Algorithm 5 has the same behaviour shown in the previous benchmark case. In particular, this inverse solver confirms to be badly affected by $p_g > 0 \frac{K^2}{W^2}$. The effect of p_g on its performance is very non linear with a first region of no effects for $p_g \lesssim 1e - 11 \frac{K^2}{W^2}$ followed by a step degradation and a plateau at 100% relative error for $p_g \gtrsim 1e - 7 \frac{K^2}{W^2}$. Moreover, the different meshes and Δt have the same behaviour and similar values of the relative error. These results confirm once more the almost insensibility of this inverse solver

to the discretization refinement.

Now, we consider the piecewise linear approximation of Algorithm 6. For this inverse solver, Figures 9.3.7 and 9.3.8 show the mean and maximum values of the L^2 -norm of the relative error, e_{rel} , in the interval $(0, t_f]$ as functions of p_g changing the mesh and the Δt , respectively.

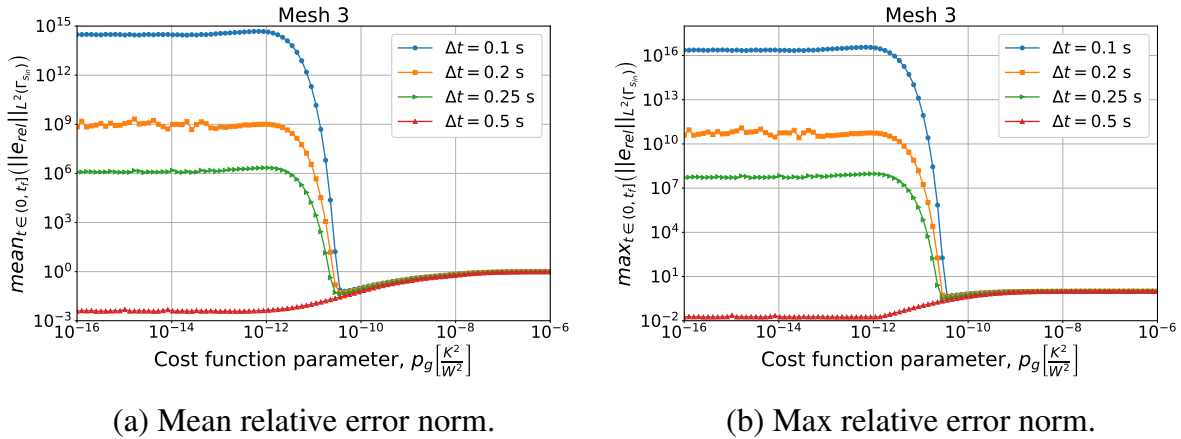


Figure 9.3.7: Mean and maximum values of the L^2 -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, for Benchmark 3 as the value of the cost function parameter, p_g , changes for Algorithm 6 (piecewise linear time approximation of the heat flux). We show the results for Mesh 3 and different Δt .

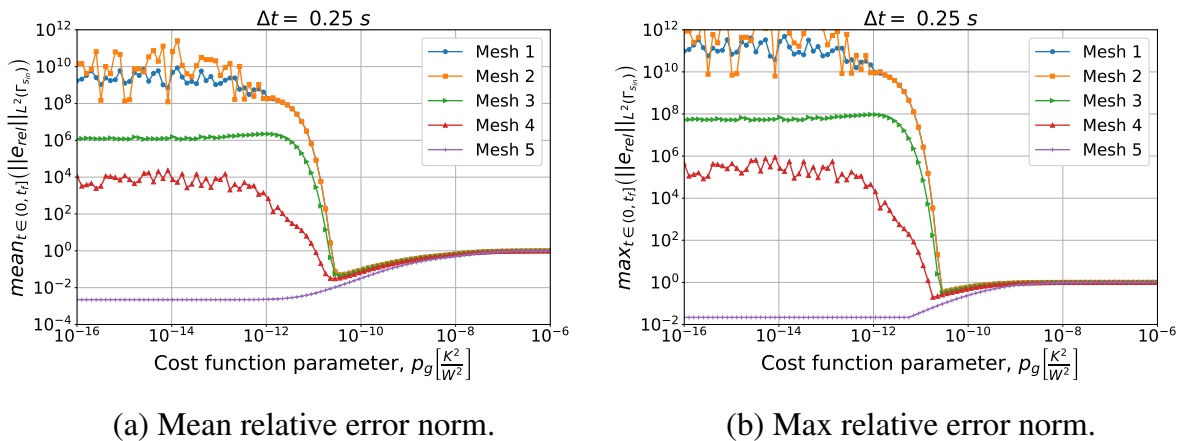


Figure 9.3.8: Mean and maximum values of the L^2 -norm of the relative error, e_{rel} , in the interval $(0, t_f]$, for Benchmark 3 as the value of the cost function parameter, p_g , changes. The results are obtained using Algorithm 6 (piecewise linear time approximation of the heat flux). We show the results for $\Delta t = 0.25s$ and different meshes.

Again, the results are very similar to those of the previous benchmark. On one hand, the coarsest discretizations show a similar behaviour to the piecewise constant

approximation case with a monotonic degradation of the performance as p_g increases. On the other hand, we have unstable solutions for small p_g that are stabilized by $p_g \gtrsim 5e - 11 \frac{K^2}{W^2}$. However, the accuracy of these solution rapidly decreases as we further increase p_g until we reach the 100% relative error plateau.

To conclude, we test the mesh, Δt and p_g selection method of Algorithm 7. In this test, we input to the algorithm the virtual thermocouples measurements that we compute for this benchmark case. Before presenting the results of this algorithm, we show in Figure 9.3.9 the mean value of S_1^k , m_S , as function of p_g for different meshes and Δt .

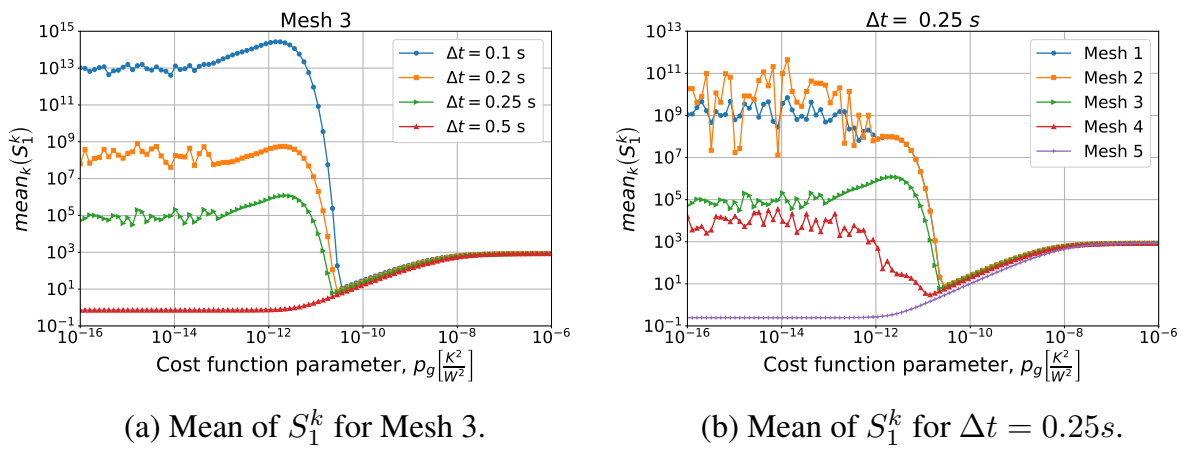


Figure 9.3.9: Mean values of S_1^k for $1 \leq k \leq P_t$, for Benchmark 3 as the value of the cost function parameter, p_g , changes. The results are obtained using Algorithm 6 (piecewise linear time approximation of the heat flux). We show the results for Mesh 3 and different Δt in (a), and for $\Delta t = 0.25$ and different meshes in (b).

Notice that m_S has a behaviour that is very similar to the relative error norm shown in Figures 9.3.7 and 9.3.8. Moreover, its minima are almost correspondent to the relative error ones. As already mentioned, we used these results as a guideline in developing Algorithm 7. We present its behaviour for the present benchmark case in Table 9.9.

Table 9.9: Test of Algorithm 7 in the unsteady Benchmark 3.

Iteration	Mesh	Δt [s]	$p_g \left[\frac{K^2}{W^2} \right]$	$mean_k(S_1^k) [K^2]$
0	4	0.1	$8e - 7$	$8.46e2$
1	5	0.5	$3.6e - 11$	$6.2e0$
2	5	0.5	$3.2e - 13$	$5.5e - 1$

In this case, the algorithm does not select since the beginning the coarsest dis-

cretization. In the first iteration, it finds the p_g that minimizes m_S for this setup. Then, when comparing it to the other discretizations in step 23, it selects again the coarsest one. Also in this benchmark case, this algorithm showed to be able to select the discretization setup and the value of p_g corresponding to the best performance of the inverse solver.

10 CONCLUSIONS AND FUTURE PROSPECTIVE

The goal of the present investigation was to develop mathematical tools to monitor the mold behaviour in CC machineries. At industrial experts' suggestion, we identified the mold-steel heat flux as the quantity of interest for the mold monitoring. Then, our objective has been to investigate a methodology for the estimation of this heat flux having as data the physical properties of the mold, its geometry, the cooling water temperature and some pointwise temperature measurements in the interior of the mold's plates.

We opted for stating the problem in a data assimilation, optimal control setting in which we look for the heat flux that minimizes a functional that includes a measure of the distance between the computed and measured temperature at the measurement points. However, given a mold-steel heat flux, we required a mold model to compute the temperature at these points.

In deriving the mold model, we had to take into account the real-time requirement of this mold monitoring task. Then, we avoided to model all the mold region, including the steel pool. In fact, it would require us to model several complex and coupled physical phenomena (heat transfer, steel solidification, steel and cooling water fluid flows, etc.) happening on different time and space scales. With this level of complexity, the computational cost of such simulation would have been unsustainable for real-time computations. Thus, in modeling the mold, we considered as computational domain the mold plates solid region and the cooling water passing through them. Hence, our first task was to develop a hierarchy of models by adding simplifying assumptions to a first general model.

Once the mold models hierarchy has been established, we focused on the mold-steel heat flux estimation. Notice that, in this setting, this flux is a Neumann BC on a portion of the boundary of our domain. Then, we can generalize this mathematical inverse problem as the estimation of a Neumann BC given pointwise state measurements in the interior of the domain.

Coming back to our application, when dealing with the inverse problem, we employed two of the developed mold models. In particular, we used a three-dimensional steady-state heat conduction model in Part I and its unsteady-state version in Part II of this thesis.

In the steady-state case, we proposed two different inverse problems and two methodologies for their solutions. The two inverse problems differ in the available measured data. Firstly, we considered to have available the pointwise temperature measurements only. Then, we assumed to measure also the total boundary heat flux. In fact, under the steady-state assumption, we can estimate the heat flux extracted from the steel by the mold by knowing the cooling water temperature increase.

As mentioned, we proposed two methodologies for the solution of these inverse problems. The first one is Alifanov's regularization which is a classic technique to solve IHTPs based on the CG method. The second method exploits a parameterization of the heat flux with RBFs. Thanks to this parameterization, we developed a direct (non-iterative) method to estimate the mold-steel heat flux.

While Alifanov's regularization is an iterative method that requires the solution of three direct problems at each iteration, the parameterization method turned out to benefit from an offline-online decomposition. Thanks to this decomposition, we have a first computationally expensive offline phase, in which we solve several direct problems. This offline phase is computed once and for all and does not require any measurements. Then, when the caster starts to work, we only have to collect the thermocouples measurements and run the online phase which is computationally much cheaper and its cost does not depend on the refinement of the used mesh.

To test the performance and the CPU requirements of these methods, we designed different numerical benchmark cases. In these tests, the parameterization method outperformed Alifanov's regularization in the quality of the heat flux reconstruction. Even more, in the tests where we reproduced the industrial setting, Alifanov's regularization had very poor performance. In fact, in CC molds the measurements are very close to the boundary where we want to reconstruct the heat flux and this badly affects Alifanov's regularization behaviour. With respect to the CPU requirements of these methods, while Alifanov's regularization cannot be applied in real-time as it is, the novel parameterization method proved to meet the real-time requirement of this application.

We also tested the robustness of these inverse solvers to the presence of noise in the measurements and their ability to filter it out. Both showed to be effective under mild noise with the parameterization method (thanks to the TSVD regularization) being able to handle also cases with severe noise. Then, the proposed parameterization method proved to meet all the requirements of this application on the quality of the heat flux estimation, the real-time performance and the robustness to errors in the measurements.

After addressing the mold modeling and the estimation of the mold-steel heat

flux with a 3D steady-state heat conduction model, in Part II of this thesis, we moved a step forward considering an unsteady-state mold model. In this new setting, we used a sequential approach to the inverse problem. In fact, to provide a real-time solution in an unsteady setting means to stay always at the front of the time line as it stretches. Then, since our measurements come equally spaced in time by one second, we considered the problem of estimating the heat flux only in between the last measurement and the previous one, assuming to have already the solution for older times.

In this framework, we stated two different inverse problems. One looking for the heat flux that minimizes a measure of the distance between computed and measured temperature only. While, in the other, we want to minimize this distance plus an heat flux norm.

For both these inverse problem, we developed novel methodologies for their solution that, mimicking the one created for the steady case, extends it to the unsteady one. In particular, these novel methods exploit the same RBFs parameterization in space of the steady-state one, but use time dependent coefficients. With respect to these coefficients, we considered both the piecewise constant and the piecewise linear case. It means that the estimated heat flux is constant or linear in between two contiguous measurement instants.

Even if this novel methodology can still exploit an offline-online decomposition, it has the drawback of requiring in the online phase the solution of one direct problem. As a consequence, this method is not suitable for real-time computation as it is. Then, we addressed the problem of developing a ROM for this direct problem to speed up its solution and meet the real-time requirement.

To construct such ROM, we used the RB POD-Galerkin method. The novelty on this regard is the methodology used for sampling the parameter space and create the POD RB. In fact, we developed a data-driven incremental algorithm that automatically enrich the RB space. It as been derived by combining incremental SVD techniques with a novel projection error estimator. Then, we proposed a novel inverse solver that includes the creation of this ROM in the offline phase and uses it in the online phase. It allowed us to have a mesh independent, real-time online phase.

To conclude, we tested the proposed inverse solvers on some benchmark cases. First, we validated our FVM simulation of the direct problem. Then, we designed two benchmark cases for the inverse problem. To design an inverse problem test, we arbitrarily selected a mold-steel heat flux. We solved the direct problem for this heat flux obtaining the corresponding temperature field in the mold. Finally, we located some virtual thermocouples and used the computed temperature at this points as input

data for the inverse solvers. Then, we tested their ability to reconstruct the previously selected heat flux.

The two benchmark cases share the same domain and physical parameters which were chosen to resemble the industrial setting. The difference is in the selected heat fluxes.

In the results, we noticed a great difference in the behaviour of the piecewise constant and linear inverse solvers. The former showed a very stable behaviour and indifference to the time and space discretization used. The latter, rather, is very much influenced by the discretization used. In particular, it can be very unstable when using fine discretizations but this instability is reduced by coarsening the time and/or space discretization. In fact, for some discretizations, we achieved very stable and accurate solutions, eventually.

We also tested the effects that adding the heat flux norm to the minimization functional has on these inverse solvers. We implemented this new term multiplying it by a parameter. Then, we tested the effect that its value has on the solvers performance.

We noticed that the piecewise constant algorithm performance monotonically deteriorates as this parameter increases. The same goes for the piecewise linear solver when using the coarsest, stable discretizations. However, the unstable configurations showed to be positively affected by the addition of this new term and, for some values of this parameter, we were able to obtain stable and accurate solutions for all the tested discretizations. While, for too high values of the parameter, the solution is stable but inaccurate for all the meshes and timestep sizes. Moreover, we showed that, for values of the parameter above a threshold, the inverse solver performance are almost independent from the discretization.

Finally, due to the great importance played by the mesh refinement, the timestep size and the heat flux norm parameter, we developed an algorithm for their automated selection. We tested it on the designed benchmark cases and it shown to be able to select stable and accurate setups. This proposed algorithm is important in the lack of theoretical results that can ensure a priori the stability and accuracy of this inverse solver.

This is surely a possible future work on this subject that would increase the potential of the propose methodology and increase its reliability. In particular, it would be useful for the final user to know a priori the time and space discretization to select as well as the minimization functional parameter. Notice that it is needed for the piecewise linear inverse solver because the piecewise constant one is almost insensible to the discretization refinement.

In the future, it would also be interesting to investigate the application of the more

sophisticated mold models that are in our models hierarchy, and to study the solution of the related inverse problems. Since they are coupled multiphase models, we expect their solution to require the development of novel techniques. Moreover, to meet the real-time requirement of this application will probably require the derivation of new MOR techniques.

Another possible future investigation on this subject could be to use a completely different approach in the solution of this inverse problem. In particular, thinking about a more proper handling of the measurement noise, we could think of using a Bayesian approach. Techniques such as ensemble Kalman filter could be suitable for this problem given the sequentiality of the measurements. Moreover, considering the real-time requirement of the application, it would probably require an effective use of MOR technique to reduce the well known computational limit of these techniques.

BIBLIOGRAPHY

- [1] AISE Steel Foundation and A. W. Cramb. *The making, shaping, and treating of steel: Casting Volume*. AISE Steel Foundation, 2003.
- [2] O. Alifanov. *Inverse Heat Transfer Problems*. Moscow Izdatel Mashinostroe-
nie, 1 edition, 1988.
- [3] D. Ambrosi. Cellular traction as an inverse problem. *SIAM Journal of Applied
Mathematics*, 66:2049–2060, 2006.
- [4] R. Aster. *Parameter estimation and inverse problems*. Elsevier, Amsterdam,
Netherlands, 2019.
- [5] I. Babuška. The finite element method for elliptic equations with discontinuous
coefficients. *Computing*, 5(3):207–213, 09 1970.
- [6] L. Bai, B. Wang, H. Zhong, J. Ni, Q. Zhai, and J. Zhang. Experimental and
numerical simulations of the solidification process in continuous casting of
slab. *Metals*, 6:53, 03 2016.
- [7] J. M. Bardsley. *Computational Uncertainty Quantification for Inverse Prob-
lems*. SIAM, 2018.
- [8] P. Barral, A. Bermudez, M. Muñiz, M. Otero, P. Quintela, and P. Salgado.
Numerical simulation of some problems related to aluminium casting. *Journal
of Materials Processing Technology*, 142:383–399, 11 2003.
- [9] J. Beck. Nonlinear estimation applied to the nonlinear inverse heat conduction
problem. *International Journal of Heat and Mass Transfer*, 13(4):703–716,
1970.
- [10] J. V. Beck. Surface heat flux determination using an integral method. *Nuclear
Engineering and Design*, 7(2):170 – 178, 1968.
- [11] J. V. Beck, B. Blackwell, and C. R. S. Clair Jr. *Inverse heat conduction: Ill-
posed problems*. James Beck, 1985.

-
- [12] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. Miguel Silveira. *Model Order Reduction. Volume 1: System-and Data-Driven Methods and Algorithms*. De Gruyter, 2020.
- [13] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. Miguel Silveira. *Model Order Reduction. Volume 2: Snapshot-Based Methods and Algorithms*. De Gruyter, 12 2020.
- [14] P. Benner, R. Herzog, N. Lang, I. Riedel, and J. Saak. Comparison of model order reduction methods for optimal sensor placement for thermo-elastic models. *Engineering Optimization*, 51(3):465–483, 2019.
- [15] P. Benner, E. Sachs, and S. Volkwein. *Trends in PDE Constrained Optimization*, chapter Model Order Reduction for PDE Constrained Optimization, pages 303–326. Springer International Publishing, Cham, 2014.
- [16] P. Benner and J. Schneider. Uncertainty quantification for maxwell’s equations using stochastic collocation and model order reduction. *International Journal for Uncertainty Quantification*, 5(3):195–208, 2015.
- [17] A. Bermúdez, M. Muñiz, and P. Quintela. Existence of solution for a free boundary problem in a nonlinear piecewise homogenous medium. *Annales de l’Institut Henri Poincaré-Nonlinear Analysis*, 15(4):399–430, 1998.
- [18] M. Bernreuther, G. Müller, and S. Volkwein. Reduced basis model order reduction in optimal control of a nonsmooth semilinear elliptic pde. In *Optimization and Control for Partial Differential Equations*, Radon Series on Computational and Applied Mathematics. De Gruyter, 2022.
- [19] F. E. Bock, R. C. Aydin, C. J. Cyron, N. Huber, S. R. Kalidindi, and B. Klusemann. A review of the application of machine learning and data mining approaches in continuum materials mechanics. *Frontiers in Materials*, 6, 05 2019.
- [20] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Computer Vision — ECCV 2002*, pages 707–720. Springer Berlin Heidelberg, 2002.
- [21] B. Brands, J. Mergheim, and P. Steinmann. Reduced-order modelling for linear heat conduction with parametrised moving heat sources. *GAMM-Mitteilungen*, 39(2):170–188, 2016.

- [22] T. Breiten and T. Damm. Krylov subspace methods for model order reduction of bilinear control systems. *Systems & Control Letters*, 59(8):443–450, 08 2010.
- [23] M. D. Buhmann. *Radial basis functions: theory and implementations*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge university press, 2003.
- [24] M. B. Carver. Numerical Computation of Phase Separation in Two Fluid Flow. *Journal of Fluids Engineering*, 106(2):147–153, 06 1984.
- [25] Y. A. Cengel. *Heat transfer: a practical approach*. Boston, Mass: WBC McGraw-Hill, 2007.
- [26] C. W. Chang, C. H. Liu, and C. C. Wang. Review of computational schemes in inverse heat conduction problems. *Smart Science*, 6(1):94–103, 2018.
- [27] L. Chang, W. Gong, and N. Yan. Numerical analysis for the approximation of optimal control problems with pointwise observations. *Mathematical Methods in the Applied Sciences*, 38:4502–4520, 2015.
- [28] S. Chaturantabut and D. C. Sorensen. Discrete empirical interpolation for nonlinear model reduction. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 4316–4321, 2009.
- [29] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 01 2010.
- [30] R. Chaudhary, C. Ji, B. G. Thomas, and S. P. Vanka. Transient turbulent flow in a liquid-metal model of continuous casting, including comparison of six different methods. *Metallurgical and Materials Transactions B*, 42(5):987–1007, 2011.
- [31] C. J. Chen and J. S. Chiou. Prediction of surface temperature and heat flux from an interior temperature response. *Letters in Heat and Mass Transfer*, 3(6):539 – 548, 1976.
- [32] H. Chen, L. Su, G. Wang, W. Shibin, L. Zhang, and Z. Luo. Fuzzy estimation for heat flux distribution at the slab continuous casting mold surface. *International Journal of Thermal Sciences*, 83:80–88, 09 2014.

-
- [33] W. L. Chen and Y. C. Yang. Inverse problem of estimating the heat flux at the roller/workpiece interface during a rolling process. *Applied Thermal Engineering*, 30(10):1247 – 1254, 2010.
- [34] F. Chinesta, A. Huerta, G. Rozza, and K. Willcox. *Encyclopedia of Computational Mechanics Second Edition*, chapter Model Reduction Methods, pages 1–36. John Wiley & Sons, Ltd, 2017.
- [35] Danieli & C. Officine Meccaniche S.p.A. https://www.danieli.com/en/_1.htm. Accessed: 2018-09-11.
- [36] L. B. Dantas and H. R. B. Orlande. A function estimation approach for determining temperature-dependent thermophysical properties. *Inverse Problems in Engineering*, 3(4):261–279, 1996.
- [37] F. Du, X. Wang, Y. Liu, T. Li, and M. Yao. Investigation on thermo-mechanical behavior of mold corner for continuous casting slab. *ISIJ International*, 55(10):2150–2157, 2015.
- [38] F. Du, X. Wang, Y. Liu, T. Li, and M. Yao. Analysis of non-uniform mechanical behavior for a continuous casting mold based on heat flux from inverse problem. *Journal of Iron and Steel Research, International*, 23(2):83 – 91, 2016.
- [39] F. Du, X. Wang, M. Yao, and X. Zhang. Analysis of the non-uniform thermal behavior in slab continuous casting mold based on the inverse finite-element model. *Journal of Materials Processing Technology*, 214(11):2676 – 2683, 2014.
- [40] J. Eason, L. Biegler, B. Lohmann, T. Bechtold, P. Eberhard, J. Fehr, D. Rixen, M. Cruz Varona, C. Lerch, C. Yuan, E. Rudnyi, B. Fröhlich, P. Holzwarth, D. Grunert, C. Meyer, J. Rutzmoser, E. Deckers, W. Desmet, and B. Haasdonk. *Model Order Reduction. Volume 3: Applications*. De Gruyter, 12 2020.
- [41] M. Efe and H. Ozbay. Proper orthogonal decomposition for reduced order modeling: 2D heat flow. In *Proceedings of 2003 IEEE Conference on Control Applications*, volume 2, pages 1273–1277, 2003.
- [42] Engineering ToolBox. Specific Heat of some Metals. https://www.engineeringtoolbox.com/specific-heat-metals-d_152.html, 2003. Accessed: 26/05/2020.

- [43] Engineering ToolBox. Water - Density, Specific Weight and Thermal Expansion Coefficient. https://www.engineeringtoolbox.com/water-density-specific-weight-d_595.html, 2003. Accessed: 26/05/2020.
- [44] Engineering ToolBox. Metals and Alloys - Densities. https://www.engineeringtoolbox.com/metal-alloys-densities-d_50.html, 2004. Accessed: 26/05/2020.
- [45] Engineering ToolBox. Water - Dynamic and Kinematic Viscosity. https://www.engineeringtoolbox.com/water-dynamic-kinematic-viscosity-d_596.html, 2004. Accessed: 26/05/2020.
- [46] Engineering ToolBox. Water - Specific Heat. https://www.engineeringtoolbox.com/specific-heat-capacity-water-d_660.html, 2004. Accessed: 26/05/2020.
- [47] Engineering ToolBox. Thermal Conductivity of Metals, Metallic Elements and Alloys. https://www.engineeringtoolbox.com/thermal-conductivity-metals-d_858.html, 2005. Accessed: 26/05/2020.
- [48] Engineering ToolBox. Water - Thermal Conductivity. <https://www.engineeringtoolbox.com/water-liquid-gas-thermal-conductivity-temperature-pressure-2012.html>, 2018. Accessed: 26/05/2020.
- [49] H. Engl. *Regularization of inverse problems*. Kluwer Academic Publishers, Dordrecht Boston, 1996.
- [50] D. Estep, S. Tavener, and T. Wildey. A posteriori error estimation and adaptive mesh refinement for a multiscale operator decomposition approach to fluid–solid heat transfer. *Journal of Computational Physics*, 229(11):4143 – 4158, 2010.
- [51] European Commission. Directorate General for Education and Culture. *Marie Skłodowska-Curie actions, 2021-2027: developing talents, advancing research*. Publications Office, 2021.

-
- [52] R. Eymard, T. Gallouët, and R. Herbin. Finite volume methods. In *Solution of Equation in R^n , Techniques of Scientific Computing (Part 3)*, volume 7 of *Handbook of Numerical Analysis*, pages 713 – 1018. Elsevier, 2000.
- [53] O. Ezvan, A. Batou, C. Soize, and L. Gagliardini. Multilevel model reduction for uncertainty quantification in computational structural dynamics. *Computational Mechanics*, 59(2):219–246, 10 2016.
- [54] H. Fareed, J. R. Singler, Y. Zhang, and J. Shen. Incremental proper orthogonal decomposition for PDE simulation data. *Computers & Mathematics with Applications*, 75(6):1942–1960, 03 2018.
- [55] H. Festjens, G. Chevallier, and J. Dion. Nonlinear model order reduction of jointed structures for dynamic analysis. *Journal of Sound and Vibration*, 333(7):2100–2113, 2014.
- [56] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, 01 1964.
- [57] B. Fröhlich, J. Gade, F. Geiger, M. Bischoff, and P. Eberhard. Geometric element parameterization and parametric model order reduction in finite element based shape optimization. *Comput. Mech.*, 63(5):853–868, 05 2019.
- [58] M. Gadalla, M. Cianferra, M. Tezzele, G. Stabile, A. Mola, and G. Rozza. On the comparison of LES data-driven reduced order approaches for hydroacoustic analysis. *Computers & Fluids*, 216:104819, 02 2021.
- [59] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International Journal for Numerical Methods in Engineering*, 81(12):1581–1608, 2010.
- [60] S. Georgaka, G. Stabile, K. Star, G. Rozza, and M. J. Bluck. A hybrid reduced order method for modelling turbulent heat transfer problems. *Computers & Fluids*, 208:104615, 08 2020.
- [61] G. H. Golub, P. C. Hansen, and D. P. O’Leary. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999.

- [62] M. González, M. Goldschmit, A. Assanelli, E. Dvorkin, and E. Berdager. Modeling of the solidification process in a continuous casting installation for steel slabs. *Metallurgical and Materials Transactions B*, 34:455–473, 08 2003.
- [63] A. Gupta, R. Singh, A. Paul, and S. Kumar. Effect of mould geometry, coating, and plate thickness on the thermal profile of continuous casting moulds. *Journal of the Southern African Institute of Mining and Metallurgy*, 118(5):505–513, 2018.
- [64] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(2):277–302, 03 2008.
- [65] P. C. Hansen. *Discrete Inverse Problems*. Society for Industrial and Applied Mathematics, 01 2010.
- [66] W. He, P. Avery, and C. Farhat. In situ adaptive reduction of nonlinear multi-scale structural dynamics models. *International Journal for Numerical Methods in Engineering*, 121(22):4971–4988, 2020.
- [67] Y. Hebi, Y. Man, and F. Dacheng. 3-d inverse problem continuous model for thermal behavior of mould process based on the temperature measurements in plant trial. *Isij International - ISIJ INT*, 46:539–545, 01 2006.
- [68] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer International Publishing, 2015.
- [69] P. Hu, X. Wang, J. Wei, M. Yao, and Q. Guo. Investigation of liquid/solid slag and air gap behavior inside the mold during continuous slab casting. *ISIJ International*, 58(5):892–898, 2018.
- [70] A. Huespe, A. Cardona, and V. Fachinotti. Thermomechanical model of a continuous casting process. *Computer Methods in Applied Mechanics and Engineering*, 182(3):439 – 455, 2000.
- [71] T. Hummel, C. Temmler, B. Schuermans, and T. Sattelmayer. Reduced-order modeling of aeroacoustic systems for stability analyses of thermoacoustically noncompact gas turbine combustors. *Journal of Engineering for Gas Turbines and Power*, 138(5), 08 2015.

-
- [72] W. R. Irving. *Continuous casting of steel*. The Institute of Materials(UK), 1993.
- [73] ITHACA-FV. <https://mathlab.sissa.it/ithaca-fv>. Accessed: 21/04/2022.
- [74] P. N. Jalali. *Analysis of Different Continuous Casting Practices Through Numerical Modelling*. PhD thesis, KTH Royal Institute of Technology, 2013.
- [75] G. Jiang, H. Liu, K. Yang, and X. Gao. A fast reduced-order model for radial integral boundary element method based on proper orthogonal decomposition in nonlinear transient heat conduction problems. *Computer Methods in Applied Mechanics and Engineering*, 368:113190, 2020.
- [76] Z. Jinqiang and X. Luo. Estimation of heat transfer coefficients and heat flux on the billet surface by an integrated approach. *International Journal of Heat and Mass Transfer*, 90:645–653, 07 2015.
- [77] H. Kania and J. Gawor. Impact of mould powder density on surface quality and near-surface zone microstructure of cast slab. *Archives of Metallurgy and Materials*, 57(1):339 – 345, 2012.
- [78] J. Kiijarvi. Darcy friction factor formulae in turbulent pipe flow. *Lunowa Fluid Mechanics Paper*, 110727:1–11, 2011.
- [79] M. P. Kirkpatrick and S. W. Armfield. Open boundary conditions in numerical simulations of unsteady incompressible flow. In G. N. Mercer and A. J. Roberts, editors, *Proceedings of the 14th Biennial Computational Techniques and Applications Conference, CTAC-2008*, volume 50 of *ANZIAM J.*, pages C760–C773, 2009.
- [80] A. Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*, chapter Regularization by Discretization, pages 63–119. Springer New York, New York, NY, 2011.
- [81] L. Klimeš and J. Štětina. A rapid GPU-based heat transfer and solidification model for dynamic computer simulations of continuous steel casting. *Journal of Materials Processing Technology*, 226:1–14, 2015.
- [82] S. Koric, L. C. Hibbeler, R. Liu, and B. G. Thomas. Multiphysics model of metal solidification on the continuum level. *Numerical Heat Transfer, Part B: Fundamentals*, 58(6):371–392, 2010.

- [83] A. Kumar and P. Dutta. Modeling of transport phenomena in continuous casting of non-dendritic billets. *International Journal of Heat and Mass Transfer*, 48:3674–3688, 08 2004.
- [84] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40(2):492–515, 01 2002.
- [85] P. Lamb, S. Yu, H. Kytonen, and S. Louhenkilpi. The effective thermal conductivity method in continuous casting of steel. *Acta Polytechnica Hungarica*, 11:5–22, 01 2014.
- [86] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. Model order reduction in fluid dynamics: Challenges and perspectives. In *Reduced Order Methods for Modeling and Computational Reduction*, pages 235–273. Springer International Publishing, 2014.
- [87] J. L. Lions. *Optimal control of systems governed by partial differential equations problèmes aux limites*. Springer, 1971.
- [88] Z. Liu and B. Li. Effect of vertical length on asymmetric flow and inclusion transport in vertical-bending continuous caster. *Powder Technology*, 323:403–415, 2018.
- [89] A. Luikov. *Heat and Mass Transfer in Capillary-Porous Bodies*. Advances in Heat Transfer. Elsevier, 1964.
- [90] R. Mahapatra, J. Brimacombe, and I. Samarasekera. Mold behavior and its influence on quality in the continuous casting of steel slabs: Part II. mold heat transfer, mold flux behavior, formation of oscillation marks, longitudinal off-corner depressions, and subsurface cracks. *Metallurgical and Materials Transactions B*, 22(6):875–888, 1991.
- [91] Y. Man, Y. Hebi, and F. Dacheng. Real-time analysis on non-uniform heat transfer and solidification in mould of continuous casting round billets. *Isij International - ISIJ INT*, 44:1696–1704, 01 2004.
- [92] M. Martinolli, F. Cornat, and C. Vergara. Computational fluid–structure interaction study of a new wave membrane blood pump. *Cardiovascular Engineering and Technology*, 11 2021.

-
- [93] Y. Meng and B. Thomas. Modeling transient slag-layer phenomena in the shell/mold gap in continuous casting of steel. *Metallurgical and Materials Transactions B*, 34:707–725, 10 2003.
- [94] S. Michelic, W. Rauter, M. Erker, W. Brandl, and C. Bernhard. Heat transfer in a round cc mould: Measurement, modelling and validation. In *6th European Continuous Casting Conference*, pages Paper–30, Italy, 2008. Steelmaking & Plastic Deformation Study Groups, Associazione Italiana di Metallurgia.
- [95] E. Monteiro, R. Almeida, and A. Roubo. Finite volume method analysis of heat transfer in multiblock grid during solidification. In *Heat Transfer - Mathematical Modelling, Numerical Methods and Information Technology*. InTech, 02 2011.
- [96] A. Moraes, P. Lage, G. Cunha, and L. Silva. Analysis of the non-orthogonality correction of finite volume discretization on unstructured meshes. In *22nd International Congress of Mechanical Engineering*, pages 3519–3530, Ribeirão Preto, SP, Brazil, 11 2013. COBEM 2013.
- [97] U. E. Morelli, P. Barral, P. Quintela, G. Rozza, and G. Stabile. A numerical approach for heat flux estimation in thin slabs continuous casting molds using data assimilation. *International Journal for Numerical Methods in Engineering*, 122(17):4541–4574, 2021.
- [98] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab*. Springer Publishing Company, Incorporated, 1st edition, 2015.
- [99] F. D. Moura Neto and A. J. da Silva Neto. *An Introduction to Inverse Problems with Applications*. Springer Publishing Company, Incorporated, 2012.
- [100] M. Muhieddine, E. Canot, and R. March. Various approaches for solving problems in heat conduction with phase change. *International Journal On Finite Volumes*, 6, 01 2009.
- [101] J. L. Nazareth. Conjugate gradient method. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(3):348–353, 2009.
- [102] J. Necas, C. Simader, Š. Necasová, G. Tronel, and A. Kufner. *Direct Methods in the Theory of Elliptic Equations*. Springer Monographs in Mathematics. Springer Berlin Heidelberg, 2012.

- [103] F. Negri, G. Rozza, A. Manzoni, and A. Quarteroni. Reduced basis method for parametrized elliptic optimal control problems. *SIAM Journal on Scientific Computing*, 35(5):A2316–A2340, 2013.
- [104] J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.
- [105] N.-C. Nguyen, G. Rozza, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous burgers’ equation. *Calcolo*, 46(3):157–185, 06 2009.
- [106] S. Niroomandi, I. Alfaro, E. Cueto, and F. Chinesta. Model order reduction for hyperelastic materials. *International Journal for Numerical Methods in Engineering*, 81(9):1180–1206, 2010.
- [107] R. Nittka. Regularity of solutions of linear second order elliptic and parabolic boundary value problems on lipschitz domains. *Journal of Differential Equations*, 251(4):860 – 880, 2011.
- [108] R. Nittka. Inhomogeneous parabolic neumann problems. *Czechoslovak Mathematical Journal*, 64(3):703–742, 2014.
- [109] M. Nonino, F. Ballarin, G. Rozza, and Y. Maday. Overcoming slowly decaying kolmogorov n-width by transport maps: application to model order reduction of fluid dynamics and fluid–structure interaction problems, 2019. arXiv:1911.06598.
- [110] D. M. Olsson and L. S. Nelson. The nelder-mead simplex procedure for function minimization. *Technometrics*, 17(1):45–51, 1975.
- [111] H. R. B. Orlande. Inverse Problems in Heat Transfer: New Trends on Solution Methodologies and Applications. *Journal of Heat Transfer*, 134(3):1–13, 01 2012. 031011.
- [112] P. Oskman, S. Yu, H. Kytonen, and S. Louhenkilpi. The effective thermal conductivity method in continuous casting of steel. *Acta Polytechnica Hungarica*, 11:5–22, 01 2014.
- [113] G. M. Oxberry, T. Kostova-Vassilevska, W. Arrighi, and K. Chand. Limited-memory adaptive snapshot selection for proper orthogonal decomposition. *International Journal for Numerical Methods in Engineering*, 109(2):198–217, 07 2016.

-
- [114] J. K. Park, B. G. Thomas, G. Li, A. Moitra, and D. Habing. Thermal and mechanical behavior of copper molds during thinslab casting (I): Plant trial and mathematical modeling. *Metallurgical and Materials Transactions B*, pages 425–436, 2002.
- [115] B. Peherstorfer and K. Willcox. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291:21–41, 07 2015.
- [116] F. Pichi, A. Quaini, and G. Rozza. A reduced order modeling technique to study bifurcating phenomena: Application to the gross-pitaevskii equation. *SIAM Journal on Scientific Computing*, 42(5):B1115–B1135, 01 2020.
- [117] F. Pichi and G. Rozza. Reduced basis approaches for parametrized bifurcation problems held by non-linear Von Kármán equations. *Journal of Scientific Computing*, 81(1):112–135, 07 2019.
- [118] F. Pichi, M. Strazzullo, F. Ballarin, and G. Rozza. Driving bifurcating parametrized nonlinear pdes by optimal control strategies: application to Navier-Stokes equations with model order reduction, 2020.
- [119] C. Pinheiro, I. Samarasekera, J. Brimacombe, and B. Walker. Mould heat transfer and continuously cast billet quality with mould flux lubrication Part 1 Mould heat transfer. *Ironmaking & Steelmaking*, 27(1):37–54, 2000.
- [120] G. Prando. *Non-Parametric Bayesian Methods for Linear System Identification*. PhD thesis, Università di Padova, 2016.
- [121] P. J. Prescott and F. P. Incropera. Modeling of dendritic solidification systems: reassessment of the continuum momentum equation. *International Journal of Heat and Mass Transfer*, 34(9):2351 – 2359, 1991.
- [122] P. J. Prescott and F. P. Incropera. The Effect of Turbulence on Solidification of a Binary Metal Alloy With Electromagnetic Stirring. *Journal of Heat Transfer*, 117(3):716–724, 08 1995.
- [123] J. Pujol. The solution of nonlinear inverse problems and the levenberg-marquardt method. *GEOPHYSICS*, 72(4):W1–W16, 2007.
- [124] A. Quarteroni. *Numerical Models for Differential Problems*. Springer Publishing Company, Incorporated, 2nd edition, 2013.

-
- [125] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations*. Springer International Publishing, 2016.
- [126] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1:3, 2011. MATHICSE report 02.2011.
- [127] P. Ranut. *Optimization and Inverse Problems in Heat Transfer*. PhD thesis, Università degli Studi di Udine, 2012.
- [128] P. Ranut, C. Persi, E. Nobile, and S. Spagnul. Estimation of Heat Flux Distribution in a Continuous Casting Mould by Inverse Heat Transfer Algorithms. In *Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, volume Volume 2: 31st Computers and Information in Engineering Conference, Parts A and B, pages 389–398. The American Society of Mechanical Engineers, 08 2011.
- [129] J.-P. Raymond. Optimal control of partial differential equations. *Université Paul Sabatier, Internet*, 2013.
- [130] D. Rempfer. On boundary conditions for incompressible Navier-Stokes problems. *Applied Mechanics Reviews May*, 59(3):107–125, 2006.
- [131] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):229–275, 05 2008.
- [132] F. Salmoiraghi, A. Scardigli, H. Telib, and G. Rozza. Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation. *International Journal of Computational Fluid Dynamics*, 32(4-5):233–247, 2018.
- [133] I. Samarasekera and J. Brimacombe. The influence of mold behavior on the production of continuously cast steel billets. *Metallurgical Transactions B*, 13(1):105–116, 1982.
- [134] I. Samarasekera and J. K. Brimacombe. The thermal field in continuous-casting moulds. *Canadian Metallurgical Quarterly*, 18(3):251–266, 1979.

-
- [135] A. Sancarlos, E. Cueto, F. Chinesta, and J.-L. Duval. A novel sparse reduced order formulation for modeling electromagnetic forces in electric motors. *SN Applied Sciences*, 3(3), 02 2021.
- [136] Y. Sato, F. Campelo, and H. Igarashi. Fast shape optimization of antennas using model order reduction. *IEEE Transactions on Magnetics*, 51(3):1–4, 2015.
- [137] L. Scholz. ROMSOC website. <https://www.romsoc.eu/>, 2018. Accessed: 30/04/2022.
- [138] Scuola Internazionale Superiore di Studi Avanzati. SISSA MathLab website. <https://mathlab.sissa.it/>. Accessed: 2022-04-30.
- [139] Scuola Internazionale Superiore di Studi Avanzati. SISSA website. <https://www.sissa.it/>. Accessed: 30/04/2022.
- [140] K.-H. Seid, G. Gilka, R. Leung, and F. Thiele. A comparison study of reduced order models for aeroacoustic applications. In *18th AIAA/CEAS Aeroacoustics Conference (33rd AIAA Aeroacoustics Conference)*. American Institute of Aeronautics and Astronautics, 06 2012.
- [141] L. Sirovich. Turbulence and the dynamics of coherent structures part I: Coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, 1987.
- [142] G. Stabile, S. Hijazi, A. Mola, S. Lorenzi, and G. Rozza. POD-Galerkin reduced order methods for CFD using finite volume discretisation: vortex shedding around a circular cylinder. *Communications in Applied and Industrial Mathematics*, 8(1):210–236, (2017).
- [143] G. Stabile and G. Rozza. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations. *Computers & Fluids*, 173:273–284, 09 2018.
- [144] G. Stabile, M. Zancanaro, and G. Rozza. Efficient geometrical parametrization for finite-volume-based reduced order methods. *International Journal for Numerical Methods in Engineering*, 121(12):2655–2682, 02 2020.
- [145] B. Stadler and R. Ramlau. Performance of an iterative wavelet reconstructor for the multi-conjugate adaptive optics RelaY of the extremely large telescope. *Journal of Astronomical Telescopes, Instruments, and Systems*, 8(02), 03 2022.

- [146] D. M. Stefanescu. *Science and engineering of casting solidification*. Springer, 2015.
- [147] J. Stetina, F. Kavička, J. Katolický, T. Mauder, and L. Klimes. Importance of the experimental investigation of a concasting technology. *MATEC Web of Conferences*, 168:07009, 01 2018.
- [148] M. Strazzullo, F. Ballarin, R. Mosetti, and G. Rozza. Model reduction for parametrized optimal control problems in environmental marine sciences and engineering. *SIAM Journal on Scientific Computing*, 40(4):B1055–B1079, 01 2018.
- [149] M. Strazzullo, F. Ballarin, and G. Rozza. POD-Galerkin model order reduction for parametrized nonlinear time-dependent optimal flow control: an application to shallow water equations. *Journal of Numerical Mathematics*, 30(1):63–84, 2022.
- [150] M. Strazzullo, Z. Zainib, F. Ballarin, and G. Rozza. Reduced order methods for parametrized non-linear and time dependent optimal flow control problems, towards applications in biomedical and environmental sciences. In F. J. Vermolen and C. Vuik, editors, *Numerical Mathematics and Advanced Applications ENUMATH 2019*, pages 841–850, Cham, 2021. Springer International Publishing.
- [151] J. Svensson, P. E. R. López, P. N. Jalali, and M. Cervantes. One-way coupling of an advanced CFD multi-physics model to FEA for predicting stress-strain in the solidifying shell during continuous casting of steel. *IOP Conference Series: Materials Science and Engineering*, 84:012–097, 06 2015.
- [152] L. Tang, M. Yao, X. Wang, and X. Zhang. Non-uniform thermal behavior and shell growth within mould for wide and thick slab continuous casting. *Steel Research International*, 83(12):1203–1213, 2012.
- [153] B. Thomas. Modeling of the continuous casting of steel - past, present, and future. *Metallurgical and Materials Transactions B*, 33:795–812, 12 2002.
- [154] B. G. Thomas. Review on modeling and simulation of continuous casting. *Steel Research Int.*, 89(1):1700312, 2018.
- [155] B. G. Thomas and F. M. Najjar. Finite element modelling of turbulent fluid flow and heat transfer in continuous casting. *Applied Mathematical Modelling*, 15(5):226 – 243, 1991.

-
- [156] B. G. Thomas, Q. Yuan, B. Zhao, and S. P. Vanka. Transient fluid-flow phenomena in the continuous steel-slab casting mold and defect formation. *JOM-e*, 58:16–36, 2006.
- [157] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.
- [158] Udayraj, S. Chakraborty, S. Ganguly, E. Chacko, S. Ajmani, and P. Talukdar. Estimation of surface heat flux in continuous casting mould with limited measurement of temperature. *International Journal of Thermal Sciences*, 118:435 – 447, 2017.
- [159] G. Vitale, L. Preziosi, and D. Ambrosi. Force traction microscopy: An inverse problem with pointwise observations. *Journal of Mathematical Analysis and Applications*, 395(2):788 – 801, 2012.
- [160] V. Voller, A. Brent, and C. Prakash. The modelling of heat, mass and solute transport in solidification systems. *International Journal of Heat and Mass Transfer*, 32(9):1719 – 1731, 1989.
- [161] W. Wang, C. Zhu, and L. Zhou. Initial solidification and its related heat transfer phenomena in the continuous casting mold. *Steel Research International*, 88(10), 2017.
- [162] X. Wang, L. Kong, F. Du, M. Yao, X. Zhang, H. Ma, and Z. Wang. Mathematical modeling of thermal resistances of mold flux and air gap in continuous casting mold based on an inverse problem. *ISIJ International*, 56:803–811, 05 2016.
- [163] X. Wang and M. Yao. Neural networks for solving the inverse heat transfer problem of continuous casting mould. In *2011 Seventh International Conference on Natural Computation*, volume 2, pages 791–794, July 2011.
- [164] F. M. White. *Fluid Mechanics*. McGraw Hill, 2011.
- [165] World Steel Association. World steel in figures 2018. *World Steel Association: Brussels, Belgium*, 2018.
- [166] M. Wu, A. Vakhrushev, G. Nummer, C. Mugrauer, A. Kharicha, and A. Ludwig. Importance of melt flow in solidifying mushy zone. *The Open Transport Phenomena Journal*, 2:16–23, 05 2010.

- [167] D. Xiao, P. Yang, F. Fang, J. Xiang, C. Pain, and I. Navon. Non-intrusive reduced order modelling of fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 303:35–54, 2016.
- [168] X. Xie, D. Chen, H. Long, M. Long, and K. Lv. Mathematical modeling of heat transfer in mold copper coupled with cooling water during the slab continuous casting process. *Metallurgical and Materials Transactions B*, 45(6):2442–2452, 12 2014.
- [169] J. Yang, Z. Cai, and M. Zhu. Transient thermo-fluid and solidification behaviors in continuous casting mold: Evolution phenomena. *ISIJ International*, 58(2):299–308, 2018.
- [170] Q. Yuan, B. Zhao, S. P. Vanka, and B. G. Thomas. Study of computational issues in simulation of transient flow in continuous casting. *Steel Research International*, 76(1):33–43, 2005.
- [171] M. J. Zahr, P. Avery, and C. Farhat. A multilevel projection-based model order reduction framework for nonlinear dynamic multiscale problems in structural and solid mechanics. *International Journal for Numerical Methods in Engineering*, 112(8):855–881, 2017.
- [172] M. J. Zahr and C. Farhat. Progressive construction of a parametric reduced-order model for PDE-constrained optimization. *International Journal for Numerical Methods in Engineering*, 102(5):1111–1135, 12 2014.
- [173] M. Zancanaro, M. Mrosek, G. Stabile, C. Othmer, and G. Rozza. Hybrid neural network reduced order modelling for turbulent flows with geometric parameters. *Fluids*, 6(8), 2021.
- [174] H. Zhang and W. Wang. Mold simulator study of heat transfer phenomenon during the initial solidification in continuous casting mold. *Metallurgical and Materials Transactions B*, 48, 01 2017.
- [175] L. Zhang, Y.-M. Rong, H.-F. Shen, and T.-Y. Huang. Solidification modeling in continuous casting by finite point method. *Journal of Materials Processing Technology*, 192-193:511 – 517, 2007. The Seventh Asia Pacific Conference on Materials Processing (7th APCMP 2006).
- [176] X. Zhang, W. Chen, Y. Ren, and L. Zhang. Mathematical modeling on the influence of casting parameters on initial solidification at the meniscus of slab

continuous casting. *Metallurgical and Materials Transactions B*, 50(3):1444–1460, 2019.

- [177] X.-b. Zhang, W. Chen, and L.-f. Zhang. A coupled model on fluid flow, heat transfer and solidification in continuous casting mold. *China Foundry*, 14(5):416–420, 2017.

A ANNEX

The present thesis includes a partial reproduction of the following publication

- U. E. Morelli^{1,2,3}, P. Barral^{1,2}, P. Quintela^{1,2}, G. Rozza³, and G. Stabile³. A numerical approach for heat flux estimation in thin slabs continuous casting molds using data assimilation. *International Journal for Numerical Methods in Engineering*, 122(17):4541–4574, 2021.

The *International Journal for Numerical Methods in Engineering* is ranked Q1 in Engineering, Applied Mathematics and Numerical Analysis. Moreover, it has an H-index of 168 and an impact factor of 3.477.

The author, Umberto Emil Morelli, had the main role in the production of the materials that form this publication. In particular, he developed the numerical methodologies, made their implementation in ITHACA-FV and wrote the article.

¹Departamento de Matemática Aplicada and Instituto de Matemáticas, Universidade de Santiago de Compostela, c/ Lope Gómez de Marzoa s/n, 15782 Santiago de Compostela, Spain

²Instituto Tecnológico de Matemática Industrial (ITMATI), Rúa Constantino Candeira s/n, 15782 Santiago de Compostela, Spain

³Scuola Internazionale Superiore di Studi Avanzati (SISSA), Via Bonomea, 265, 34136 Trieste, Italy

JOHN WILEY AND SONS LICENSE
TERMS AND CONDITIONS

Sep 12, 2022

This Agreement between Mr. Umberto Emil Morelli ("You") and John Wiley and Sons ("John Wiley and Sons") consists of your license details and the terms and conditions provided by John Wiley and Sons and Copyright Clearance Center.

License Number	5301800734643
License date	May 04, 2022
Licensed Content Publisher	John Wiley and Sons
Licensed Content Publication	International Journal for Numerical Methods in Engineering
Licensed Content Title	A numerical approach for heat flux estimation in thin slabs continuous casting molds using data assimilation
Licensed Content Author	Umberto Emil Morelli, Patricia Barral, Peregrina Quintela, et al
Licensed Content Date	Jun 2, 2021
Licensed Content Volume	122
Licensed Content Issue	17
Licensed Content Pages	34



12/09/2022, 09:47

Type of use	Dissertation/Thesis
Requestor type	Author of this Wiley article
Format	Print and electronic
Portion	Full article
Will you be translating?	No
Title	Efficient computational strategies for the control process of continuous casting machines
Institution name	Universidade de Santiago de Compostela
Expected presentation date	May 2022
Requestor Location	Mr. Umberto Emil Morelli via lavagna 3 Cairo Montenotte, SV 17014 Italy Attn: Mr. Umberto Emil Morelli
Publisher Tax ID	EU826007151
Total	0.00 EUR
Terms and Conditions	

TERMS AND CONDITIONS

This copyrighted material is owned by or exclusively licensed to John Wiley & Sons, Inc. or one of its group companies (each a "Wiley Company") or handled on behalf of a society with which a Wiley Company has exclusive publishing rights in relation to a particular work (collectively "WILEY"). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the billing and



12/09/2022, 09:47

payment terms and conditions established by the Copyright Clearance Center Inc., ("CCC's Billing and Payment terms and conditions"), at the time that you opened your RightsLink account (these are available at any time at <http://myaccount.copyright.com>).

Terms and Conditions

- The materials you have requested permission to reproduce or reuse (the "Wiley Materials") are protected by copyright.
- You are hereby granted a personal, non-exclusive, non-sub licensable (on a stand-alone basis), non-transferable, worldwide, limited license to reproduce the Wiley Materials for the purpose specified in the licensing process. This license, **and any CONTENT (PDF or image file) purchased as part of your order**, is for a one-time use only and limited to any maximum distribution number specified in the license. The first instance of republication or reuse granted by this license must be completed within two years of the date of the grant of this license (although copies prepared before the end date may be distributed thereafter). The Wiley Materials shall not be used in any other manner or for any other purpose, beyond what is granted in the license. Permission is granted subject to an appropriate acknowledgement given to the author, title of the material/book/journal and the publisher. You shall also duplicate the copyright notice that appears in the Wiley publication in your use of the Wiley Material. Permission is also granted on the understanding that nowhere in the text is a previously published source acknowledged for all or part of this Wiley Material. Any third party content is expressly excluded from this permission.
- With respect to the Wiley Materials, all rights are reserved. Except as expressly granted by the terms of the license, no part of the Wiley Materials may be copied, modified, adapted (except for minor reformatting required by the new Publication), translated, reproduced, transferred or distributed, in any form or by any means, and no derivative works may be made based on the Wiley Materials without the prior permission of the respective copyright owner. **For STM Signatory Publishers clearing permission under the terms of the [STM Permissions Guidelines](#) only, the terms of the license are extended to include subsequent editions and for editions in other languages, provided such editions are for the work as a whole in situ and does not involve the separate exploitation of the permitted figures or extracts**, You may not alter, remove or suppress in any manner any copyright, trademark or other notices displayed by the Wiley Materials. You may not license, rent, sell, loan, lease, pledge, offer as security, transfer or assign the Wiley Materials on a stand-alone basis, or any of the rights granted to you hereunder to any other person.
- The Wiley Materials and all of the intellectual property rights therein shall at all times remain the exclusive property of John



12/09/2022, 09:47

Wiley & Sons Inc, the Wiley Companies, or their respective licensors, and your interest therein is only that of having possession of and the right to reproduce the Wiley Materials pursuant to Section 2 herein during the continuance of this Agreement. You agree that you own no right, title or interest in or to the Wiley Materials or any of the intellectual property rights therein. You shall have no rights hereunder other than the license as provided for above in Section 2. No right, license or interest to any trademark, trade name, service mark or other branding ("Marks") of WILEY or its licensors is granted hereunder, and you agree that you shall not assert any such right, license or interest with respect thereto

- NEITHER WILEY NOR ITS LICENSORS MAKES ANY WARRANTY OR REPRESENTATION OF ANY KIND TO YOU OR ANY THIRD PARTY, EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO THE MATERIALS OR THE ACCURACY OF ANY INFORMATION CONTAINED IN THE MATERIALS, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, ACCURACY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, USABILITY, INTEGRATION OR NON-INFRINGEMENT AND ALL SUCH WARRANTIES ARE HEREBY EXCLUDED BY WILEY AND ITS LICENSORS AND WAIVED BY YOU.
- WILEY shall have the right to terminate this Agreement immediately upon breach of this Agreement by you.
- You shall indemnify, defend and hold harmless WILEY, its Licensors and their respective directors, officers, agents and employees, from and against any actual or threatened claims, demands, causes of action or proceedings arising from any breach of this Agreement by you.
- IN NO EVENT SHALL WILEY OR ITS LICENSORS BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR ENTITY FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, EXEMPLARY OR PUNITIVE DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, PROVISIONING, VIEWING OR USE OF THE MATERIALS REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.
- Should any provision of this Agreement be held by a court of competent jurisdiction to be illegal, invalid, or unenforceable, that

provision shall be deemed amended to achieve as nearly as possible the same economic effect as the original provision, and the legality, validity and enforceability of the remaining provisions of this Agreement shall not be affected or impaired thereby.

- The failure of either party to enforce any term or condition of this Agreement shall not constitute a waiver of either party's right to enforce each and every term and condition of this Agreement. No breach under this agreement shall be deemed waived or excused by either party unless such waiver or consent is in writing signed by the party granting such waiver or consent. The waiver by or consent of a party to a breach of any provision of this Agreement shall not operate or be construed as a waiver of or consent to any other or subsequent breach by such other party.
- This Agreement may not be assigned (including by operation of law or otherwise) by you without WILEY's prior written consent.
- Any fee required for this permission shall be non-refundable after thirty (30) days from receipt by the CCC.
- These terms and conditions together with CCC's Billing and Payment terms and conditions (which are incorporated herein) form the entire agreement between you and WILEY concerning this licensing transaction and (in the absence of fraud) supersedes all prior agreements and representations of the parties, oral or written. This Agreement may not be amended except in writing signed by both parties. This Agreement shall be binding upon and inure to the benefit of the parties' successors, legal representatives, and authorized assigns.
- In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall prevail.
- WILEY expressly reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.
- This Agreement will be void if the Type of Use, Format, Circulation, or Requestor Type was misrepresented during the licensing process.
- This Agreement shall be governed by and construed in accordance with the laws of the State of New York, USA, without regards to such state's conflict of law rules. Any legal action, suit or proceeding arising out of or relating to these Terms and Conditions or the breach thereof shall be instituted in a court of competent jurisdiction in New York County in the State of New York in the United States of America and each party hereby consents and submits to the personal jurisdiction of such court, waives any

objection to venue in such court and consents to service of process by registered or certified mail, return receipt requested, at the last known address of such party.

WILEY OPEN ACCESS TERMS AND CONDITIONS

Wiley Publishes Open Access Articles in fully Open Access Journals and in Subscription journals offering Online Open. Although most of the fully Open Access journals publish open access articles under the terms of the Creative Commons Attribution (CC BY) License only, the subscription journals and a few of the Open Access Journals offer a choice of Creative Commons Licenses. The license type is clearly identified on the article.

The Creative Commons Attribution License

The [Creative Commons Attribution License \(CC-BY\)](#) allows users to copy, distribute and transmit an article, adapt the article and make commercial use of the article. The CC-BY license permits commercial and non-

Creative Commons Attribution Non-Commercial License

The [Creative Commons Attribution Non-Commercial \(CC-BY-NC\)License](#) permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.(see below)

Creative Commons Attribution-Non-Commercial-NoDerivs License

The [Creative Commons Attribution Non-Commercial-NoDerivs License \(CC-BY-NC-ND\)](#) permits use, distribution and reproduction in any medium, provided the original work is properly cited, is not used for commercial purposes and no modifications or adaptations are made. (see below)

Use by commercial "for-profit" organizations

Use of Wiley Open Access articles for commercial, promotional, or marketing purposes requires further explicit permission from Wiley and will be subject to a fee.

Further details can be found on Wiley Online Library
<http://olabout.wiley.com/WileyCDA/Section/id-410895.html>

Other Terms and Conditions:

v1.10 Last updated September 2015





In continuous casting machineries, monitoring the mold is essential for the safety and quality of the process. Then, the objective of this thesis is to develop mathematical tools for the real-time estimation of the mold-steel heat flux which is the quantity of interest when controlling the mold behaviour. To achieve this goal, we first address the mold modelling by developing a hierarchy of models. Then, we investigate the mold-steel heat flux estimation problem in a data assimilation, optimal control framework. For its solution, we study novel methodologies that exploit the heat flux parametrization and benefit from an offline-online decomposition. To conclude, we develop cutting-edge model order reduction techniques to meet the real-time requirement of this application.