



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

ACTA DE DISERTACIÓN PÚBLICA

No. 00044

Matrícula: 2131802109

METODOS VARIACIONALES,
OPTIMIZACIÓN Y CONTROL:
APLICACIÓN A LA RECUPERACIÓN
DE CAMPOS VECTORIALES Y
ESTABILIZACIÓN DE ARREGLOS
DE MEMORIA DE JUNTAS DE
JOSEPHSON.

En la Ciudad de México, se presentaron a las 12:00 horas del día 20 del mes de septiembre del año 2016 en la Unidad Iztapalapa de la Universidad Autónoma Metropolitana, los suscritos miembros del jurado:

DR. PEDRO GONZALEZ CASANOVA HENRIQUEZ
DR. LORENZO HECTOR JUAREZ VALENCIA
DR. JULIO ERNESTO SOLIS DAUN
DRA. MARIA LUISA SANDOVAL SOLIS

Bajo la Presidencia del primero y con carácter de Secretaria la última, se reunieron a la presentación de la Disertación Pública cuya denominación aparece al margen, para la obtención del grado de:

DOCTOR EN CIENCIAS (MATEMATICAS)

DE: JORGE LOPEZ LOPEZ

y de acuerdo con el artículo 78 fracción IV del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana, los miembros del jurado resolvieron:

APROBAR

Acto continuo, el presidente del jurado comunicó al interesado el resultado de la evaluación y, en caso aprobatorio, le fue tomada la protesta.



Jorge Lopez Lopez

JORGE LOPEZ LOPEZ
ALUMNO

REVISÓ

[Signature]
LIC. JULIO CESAR DE LARA ISASSI
DIRECTOR DE SISTEMAS ESCOLARES

DIRECTOR DE LA DIVISIÓN DE CBI

[Signature]

DR. JOSE GILBERTO CORDOBA HERRERA

PRESIDENTE

[Signature]

DR. PEDRO GONZALEZ CASANOVA
HENRIQUEZ

VOCAL

[Signature]

DR. LORENZO HECTOR JUAREZ VALENCIA

VOCAL

[Signature]

DR. JULIO ERNESTO SOLIS DAUN

SECRETARIA

[Signature]

DRA. MARIA LUISA SANDOVAL SOLIS



UNIVERSIDAD AUTÓNOMA METROPOLITANA-IZTAPALAPA
DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA

Métodos variacionales, optimización y control: aplicación a la recuperación de campos vectoriales y estabilización de arreglos de memoria de juntas de Josephson

Tesis que presenta
Jorge López López
para obtener el grado de
Doctor en Ciencias (Matemáticas)

Asesor: Dr. L. Héctor Juárez Valencia

Jurado:
Presidente: Dr. Pedro González-Casanova Henriquez
Secretario: Dra. María Luisa Sandoval Solís
Vocal: Dr. Julio Ernesto Solís Daun
Vocal: Dr. L. Héctor Juárez Valencia

Cd. de México, Septiembre de 2016.

Índice

1	Introducción	1
1.1	Reconstrucción de campos vectoriales solenoidales	2
1.2	Control y estabilización de un Arreglo de Memoria de Juntas de Josephson (AMJJ)	4
2	Reconstrucción de campos vectoriales con preconditionamiento óptimo	9
2.1	Formulación matemática y el problema de punto silla	10
2.2	Un operador para el multiplicador de Lagrange	11
2.3	Algoritmo de gradiente conjugado y su preconditionamiento óptimo.	12
2.4	Discretización por medio de un método de elementos finitos mixto	15
2.5	Resultados numéricos	16
2.6	Conclusiones	23
3	Operadores de <i>Lectura/Escritura</i> para el AMJJ: Un enfoque de controlabilidad	25
3.1	Modelación matemática del AMJJ	26
3.2	El problema de minimización	26
3.3	Calculando $DJ(v)$	27
3.4	Sobre la solución del problema de minimización (3.6) cuando $\mathcal{U}_{ad} = L^2(0, T)$	29
3.4.1	Sinopsis	29
3.4.2	Aplicación del algoritmo de gradiente conjugado (A.12)-(A.19) a la solución del problema (3.6)	29
3.5	Sobre la implementación práctica del algoritmo (3.19)-(3.30) vía una aproximación por diferencias finitas del problema (3.6)	31
3.5.1	Aproximación por diferencias finitas del problema (3.6) cuando $\mathcal{U}_{ad} = L^2(0, T)$	31
3.5.2	Cálculo de la derivada $DJ^{\Delta t}(v)$	32
3.5.3	Solución por gradiente conjugado del problema de control discreto (3.31)	34
3.6	Operaciones de Memoria	36
3.6.1	Operación ‘Escribir 1’	39
3.6.2	Operación ‘Escribir 0’	43
3.6.3	Operación <i>Leer</i>	46
3.7	Alcanzando un estado de equilibrio inestable	46
3.7.1	Controlando con una junta	46
3.7.2	Controlando con dos juntas	48
3.7.3	Controlando con tres juntas	50
3.8	Conclusiones	50

4	Estabilización del AMJJ alrededor de un equilibrio inestable	51
4.1	El enfoque de controlabilidad	52
4.2	El problema de control lineal	54
4.3	Condiciones de optimalidad y solución de gradiente conjugado para el problema (4.5)	55
4.3.1	Generalidades y Sinopsis	55
4.3.2	Calculando $DJ(\mathbf{v})$	56
4.3.3	Condiciones de optimalidad para el problema (4.5)	57
4.3.4	Ecuacion funcional satisfecha por el control óptimo	58
4.3.5	Solución por gradiente conjugado del problema de control (4.5)	60
4.4	Implementación práctica del algoritmo (4.36)-(4.48) via una aproximación de diferencias finitas del problema (4.5)	62
4.4.1	Aproximación de diferencias finitas del problema (4.5) cuando $\mathcal{U} = L^2(0, T; 3)$	62
4.4.2	Condiciones de Optimalidad y solución por gradiente conjugado de (4.51) .	63
4.5	Resultados Numéricos	66
4.5.1	Intentando controlar mediante una junta	67
4.5.2	Intentando controlar con dos juntas	67
4.5.3	Controlando mediante las tres juntas	70
4.6	Conclusiones	76
5	Conclusiones Generales	77
A	Algoritmo de Gradiente Conjugado	79
A.1	Algoritmo de gradiente conjugado para funcionales cuadráticos	80
A.2	Algoritmo de gradiente conjugado para funcionales no cuadráticos en espacios de Hilbert	81
B	Método de Newton para resolver los problemas de búsqueda en línea	85
B.1	El problema de minimización continuo	86
B.2	El problema de búsqueda en línea asociado	86
B.3	Resolviendo (B.6) con el método de Newton	87
B.4	Discretización del problema (B.1)	88
B.5	El problema de búsqueda en línea asociado a (B.19)	89
B.6	Resolviendo (B.25) con el método de Newton	89

Lista de figuras

1.1	Dominio General.	3
2.1	Dominio General.	10
2.2	Elemento en \mathcal{T}_{2h} : triángulo ABC. Elementos en \mathcal{T}_h : triángulos AQP, PRC, PQR y QBR.	15
2.3	Izquierda: Campo vectorial. Derecha: Malla gruesa para el multiplicador en el ejemplo 2.	19
2.4	Vistas diferentes para el multiplicador λ en el ejemplo 2.	19
2.5	Izquierda: campos vectoriales ajustado (azul) y exacto (rojo) cerca del cilindro para el ejemplo 2. Derecha: diferencia entre los campos vectoriales exacto y ajustado (una amplificación).	20
2.6	Multiplicador λ para el ejemplo 4.	21
2.7	Campos vectoriales exacto (rojo) y ajustado (azul) para el ejemplo 4. A la izquierda se muestra una amplificación local.	22
2.8	Diferencia entre el campo exacto y el campo recobrado en el ejemplo 4 (una amplificación).	22
3.1	Los dos operadores ‘Escribir 0’ y ‘Escribir 1’ tienen que enviar el sistema al estado previsto desde cualquier estado inicial ‘0’ o ‘1’.	37
3.2	Series de tiempo de las fases escaladas por 2π , con condiciones iniciales cero.	37
3.3	Evolución al estado $\{1, 1, 0(s)\}$ desde una aproximación del equilibrio $\{1, 0, 0(u)\}$	38
3.4	Evolución al estado $\{2, 1, 0(s)\}$ desde una aproximación del equilibrio $\{2, 1, 0(u)\}$	38
3.5	Evolución al estado $\{2, 2, 0(s)\}$ desde una aproximación del equilibrio $\{2, 2, 0(u)\}$	38
3.6	Controles para varios valores de T . La transición es desde el estado ‘0’ hasta el estado ‘1’.	39
3.7	Fases de la junta 1 para diferentes valores de T , asociadas con los controles de la figura 3.6.	39
3.8	Fases de la junta 2 para diferentes valores de T , asociadas con los controles de la figura 3.6.	40
3.9	Fases de la junta 3 para diferentes valores de T , asociadas con los controles de la figura 3.6.	40
3.10	Norma relativa del estado final aproximado cuando se transita del estado ‘0’ al estado ‘1’ usando los controles de la figura 3.6.	40
3.11	Control para $T = 7$ de la figura 3.6, extendido como $u = 0$	41
3.12	Comportamiento de las fases cuando se inicia en el estado ‘0’ y se aplica en la primera junta el control de la figura 3.11. El sistema transita al estado ‘1’.	41

3.13	Comportamiento de las fases cuando se inicia en el estado ‘1’ y se aplica en la primera junta el control de la figura 3.11. El sistema transita al estado $\{2, 2, 0(s)\}$.	42
3.14	Control útil como operador ‘Escribir 1’ cuando $T = 20$.	42
3.15	Transición del sistema al estado ‘1’ desde los dos estados ‘0’ y ‘1’ usando el control de la figura 3.14.	43
3.16	Controles para varios valores de T . Controlando la tercera junta el sistema va del estado ‘1’ al estado ‘0’.	43
3.17	Fases de la junta 1 para diferentes valores de T , asociadas con los controles de la figura 3.16.	44
3.18	Fases de la junta 2 para diferentes valores de T , asociadas con los controles de la figura 3.16.	44
3.19	Fases de la junta 3 para diferentes valores de T , asociadas con los controles de la figura 3.16.	44
3.20	Norma relativa del estado final.	45
3.21	Control actuando como operador ‘Escribir 0’.	45
3.22	Transición del estado ‘1’ al estado ‘0’ (izquierda); Inicio y permanencia en el estado ‘0’ (derecha).	45
3.23	Alcanzando el equilibrio inestable $\{2, 1, 0(u)\}$ desde el estable $\{2, 1, 0(s)\}$.	47
3.24	Control aplicado en la primera junta para alcanzar el equilibrio inestable $\{2, 1, 0(u)\}$ desde el estable cercano $\{2, 1, 0(s)\}$.	47
3.25	Intento de transitar el sistema del estado ‘0’ al estado inestable $\{1, 0, 0(u)\}$, controlando en la junta 1.	47
3.26	Intento de transitar el sistema del estado ‘0’ al estado inestable $\{1, 0, 0(u)\}$, controlando en la junta 3.	48
3.27	Intento de transitar el sistema desde el estado ‘1’ hasta el estado inestable $\{1, 0, 0(u)\}$. Observese que permanece en el estado ‘1’.	48
3.28	Transición del sistema del estado estable $\{0, 0, 0(s)\}$ al estado inestable $\{1, 0, 0(u)\}$ cuando se controlan las juntas 1 y 3.	49
3.29	Transición del sistema desde el estado estable $\{1, 1, 0(s)\}$ tratando de alcanzar el estado inestable $\{1, 0, 0(u)\}$, controlando las juntas 1 y 2.	49
3.30	Transición del sistema desde el estado estable $\{1, 1, 0(s)\}$ tratando de alcanzar el estado inestable $\{1, 0, 0(u)\}$, controlando las juntas 2 y 3.	49
3.31	Transición del sistema desde el estado estable $\{1, 1, 0(s)\}$ al estado inestable $\{1, 0, 0(s)\}$ cuando se controlan las tres juntas.	50
4.1	Solución $\mathbf{y} = \delta\phi$ de (4.1) para el equilibrio inestable $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$ y $\delta\theta = [1E - 2, 1E - 2, 1E - 2]$.	52
4.2	Solución $\mathbf{y} = \delta\phi$ de (4.1) para el equilibrio inestable $\{n_1, n_2, n_3\} = \{2, 1, 0(u)\}$ y $\delta\theta = [1E - 2, 1E - 2, 1E - 2]$.	53
4.3	Solución $\mathbf{y} = \delta\phi$ de (4.1) para el equilibrio inestable $\{n_1, n_2, n_3\} = \{2, 2, 0(u)\}$ y $\delta\theta = [1E - 2, 1E - 2, 1E - 2]$.	53
4.4	Solución ϕ de (4.2) cuando no se controla y la perturbación inicial es $\delta\theta = [1E - 2, 1E - 2, 1E - 2]$ del equilibrio inestable $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$.	54
4.5	$\ \mathbf{y}^{\Delta t}(\cdot)\ $ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se intenta controlar con la primera junta.	67

4.6	$\mathbf{u}^{\Delta t}$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se intenta controlar con la primera junta.	68
4.7	$ln\ \mathbf{g}_q^{\Delta t}\ $ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se intenta controlar con la primera junta.	68
4.8	$\ \mathbf{y}^{\Delta t}(\cdot)\ $ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las juntas 1 y 2.	68
4.9	$\mathbf{u}^{\Delta t}$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las juntas 1 y 2.	69
4.10	$ln\ \mathbf{g}_q^{\Delta t}\ $ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las juntas 1 y 2.	69
4.11	$\ \mathbf{y}^{\Delta t}(\cdot)\ $ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las juntas 2 y 3.	70
4.12	$\mathbf{u}^{\Delta t}$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las juntas 2 y 3.	70
4.13	$ln\ \mathbf{g}_q^{\Delta t}\ $ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las juntas 2 y 3.	71
4.14	$\ \mathbf{y}^{\Delta t}(\cdot)\ $ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las 3 juntas.	71
4.15	$\mathbf{u}^{\Delta t}$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las 3 juntas.	71
4.16	$ln\ \mathbf{g}_q^{\Delta t}\ $ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las 3 juntas.	72
4.17	Controles para $T = 2$ cuando $k_1 = 1E + 3$ y $k_2 = 1E + 3$ y se controla con las tres juntas.	72
4.18	Norma euclidea de la solución $\mathbf{y}^{\Delta t}$ del sistema lineal cuando se usan los controles de la figura 4.17.	73
4.19	Norma euclidea de la solución $\phi^{\Delta t}$ del sistema no lineal cuando se usan los controles de la figura 4.17.	73
4.20	Componentes $y_i^{\Delta t}$ de la solución del modelo lineal perturbado, cuando se aplican los controles de la figura 4.17.	74
4.21	Componentes $\phi_i^{\Delta t}$ de la solución del modelo no lineal, cuando se aplican los controles de la figura 4.17.	74
4.22	Controles calculados iterativamente cada 2 unidades de tiempo.	75
4.23	Norma euclidiana de la solución $\mathbf{y}^{\Delta t}$ usando los controles de la figura 4.22.	75
4.24	Norma euclidiana de la solución $\phi^{\Delta t}$ usando los controles de la figura 4.22.	76

Resumen

En este trabajo de tesis se abordan dos problemas: i) Estudiamos numéricamente un modelo de diagnóstico, basado en conservación de masa, para recobrar un campo vectorial solenoidal a partir de datos experimentales incompletos. A partir de una reformulación del modelo matemático como un problema de punto silla, introducimos un algoritmo iterativo de gra-diente conjugado preconditionado, asociado a una ecuación operacional de tipo elíptico, para resolver el problema. Para obtener un algoritmo estable, usamos una aproximación de elemento finito mixto de segundo orden para la discretización. Mostramos, usando campos vectoriales sintéticos, que este nuevo enfoque, da soluciones muy precisas a un costo computacional mas bajo que los procedimientos clásicos. También, exhibimos algunas limitaciones del modelo inviscido tradicional introducido por Sasaki, mostrando que es incompatible con algunos campos vectoriales solenoidales. ii) A partir un modelo matemático de ecuaciones diferenciales ordinarias, usamos técnicas de control para manipular la transición entre estados de equili-brio estables de las fases de un circuito de tres juntas de Josephson y para estabilizar las fases del circuito alrededor de un equilibrio inestable. Para resolver computacionalmente este problema se establece también una relación funcional satisfecha por el control, lo que permite utilizar un algoritmo de gradiente conjugado. Se muestra la aplicación de la transición entre estados de equilibrio estables a la definición de operadores de *Lectura/Escritura* para memoria en computación cuántica.

Capítulo 1

Introducción

En esta tesis se abordan dos problemas modelados por ecuaciones diferenciales, en donde para resolverlos se utiliza el cálculo de variaciones, la optimización y el control, además de técnicas de discretización. Los problemas son: i) El ajuste o recuperación de campos vectoriales y ii) El control de un arreglo de memoria de juntas de Josephson (AMJJ). En ambos casos la mejor solución, o estado óptimo, resuelve un problema de optimización en espacios de Hilbert, en donde la función de costo es un funcional cuadrático convexo. Por lo tanto, se utiliza el método de gradiente conjugado en espacios de Hilbert como herramienta fundamental, además de técnicas de discretización, para encontrar soluciones efectivas en forma robusta y estable. A continuación se describe cada uno de estos problemas y algunos antecedentes.

1.1 Reconstrucción de campos vectoriales solenoidales

Varios problemas y aplicaciones requieren de un buen conocimiento de un campo vectorial sobre una región. Como por ejemplo, la predicción del transporte, difusión y dispersión de contaminantes de aire y agua en la atmósfera y en el océano [1, 2, 3]; la realización de mapas de viento para el diseño de diferentes proyectos urbanos y generales [4]. Mas aún, los campos de agua/viento meteorológicos también son datos iniciales o de entrada que se requieren para alimentar los modelos de calidad de aire/agua. Usualmente esta información está incompleta o con errores debidos a medidas experimentales, y se requiere una reconstrucción del campo vectorial. Se han propuesto varios modelos y estrategias, con diferentes niveles de complejidad, para resolver este problema. Una revisión de los modelos más usados (populares) en meteorología está disponible en [5]. Dentro de este contexto, el primer problema que estamos considerando en este trabajo es el de generar, en un dominio dado, un campo de velocidad \mathbf{u} para un fluido incompresible, el cual ajusta a uno inicial \mathbf{u}^I obtenido por medidas experimentales, por ejemplo. El campo vectorial ajustado \mathbf{u} debe satisfacer conservación de masa y una condición de *no-flujo* en parte de la frontera. Consideramos un modelo de masa consistente el cual está basado en la formulación original de Sasaki [6]. Este enfoque ha sido usado para una variedad de problemas meteorológicos [1, 4, 7, 8].

Los modelos de masa consistente son atractivos por: (i) Su simplicidad; (ii) Su facilidad de implementarse y su bajo costo de operación; (iii) Han tenido varios desarrollos y mejoras diversas durante las últimas décadas [5, 8, 9, 10, 11, 12]. La reconstrucción de campos vectoriales a partir de datos experimentales sigue siendo un tópico de investigación importante [3, 13, 14, 15], y las técnicas numéricas para resolver esos problemas en diferentes contextos son cada vez mas sofisticadas [16, 17, 18, 19], y serán más importantes conforme aumenta la complejidad de los problemas. Un tópico relacionado es la reconstrucción de un campo vectorial en cualquier punto en el espacio, dados sus valores en una vecindad del punto \mathbf{o} , mas específicamente, dada la componente normal en las aristas de la malla computacional. La reconstrucción de vectores en este caso está más relacionada a la interpolación de alto orden y puede utilizarse en diferentes contextos, como por ejemplo en los métodos de asimilación de datos, algoritmos de visualización, modelación atmosférica y costera, o su uso en esquemas semi-Lagrangianos. Se han desarrollado varios métodos de reconstrucción vectorial para resolver estos problemas para retículas planas y C-retículas esféricas poligonales. Para mas información, recomendamos las referencias [20] y [21].

Sea Ω el dominio de interés con frontera $\Gamma = \Gamma_N \cup \Gamma_D$ (ver la figura 1.1). El modelo que usamos es

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega, \tag{1.1}$$

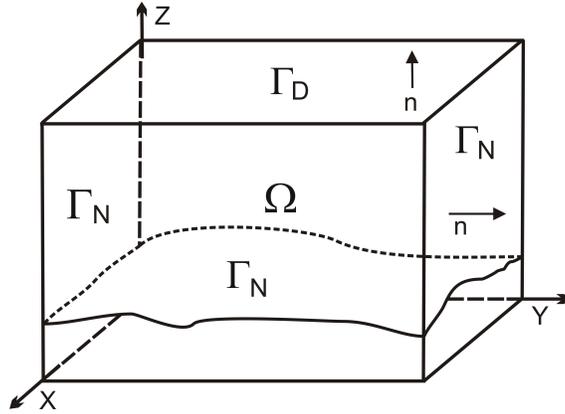


Figura 1.1: Dominio General.

$$\mathbf{u} \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_N, \quad (1.2)$$

y el problema correspondiente está definido por

$$\begin{cases} \mathbf{u} \in \mathbf{V}, \\ J(\mathbf{u}) \leq J(\mathbf{v}), \forall \mathbf{v} \in \mathbf{V}, \end{cases} \quad (1.3)$$

donde: (i) El espacio de velocidad \mathbf{V} se define como

$$\mathbf{V} = \{ \mathbf{v} \in \mathbf{H}(\text{div}; \Omega) : \nabla \cdot \mathbf{v} = 0 \text{ and } \mathbf{v} \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_N \}, \quad (1.4)$$

(ii) La función de costo J (un funcional de fidelidad, en el contexto de procesamiento de imágenes) se define como

$$J(\mathbf{v}) = \frac{1}{2} \int_{\Omega} S (\mathbf{v} - \mathbf{u}^I) \cdot (\mathbf{v} - \mathbf{u}^I) \, d\mathbf{x}, \quad (1.5)$$

En (1.4) y (1.5) (y en lo que sigue) las derivadas son en el sentido de distribuciones. El espacio $\mathbf{H}(\text{div}; \Omega)$ es definido como $\mathbf{H}(\text{div}; \Omega) = \{ \mathbf{v} \in \mathbf{L}_2(\Omega) : \nabla \cdot \mathbf{v} \in L_2(\Omega) \}$, $\hat{\mathbf{n}}$ denota el vector normal unitario exterior en Γ y S es una matriz diagonal con parámetros de peso S_{ii} , $i = 1, 2, 3$, llamados módulos de precisión Gaussiana, relacionados con las escalas de las respectivas componentes del campo de velocidad. Una aplicación en meteorología es recuperar un campo de viento a partir de datos horizontales; en este caso la componente vertical de \mathbf{u}^I se considera como cero, porque algunas estaciones meteorológicas no miden esta componente.

En algunas otras aplicaciones, como dinámica de fluidos experimental, \mathbf{u}^I puede considerarse como el resultado de una perturbación al campo vectorial original. Usando argumentos de convexidad clásicos (ver, i.e., [22]) se puede probar fácilmente que el problema asociado (1.3) tiene una única solución. Con el fin de resolver el problema (1.3) numéricamente, asociamos un multiplicador de Lagrange (que puede pensarse como un tipo de presión) con la condición $\nabla \cdot \mathbf{u} = 0$, como se hace clásicamente en fluidos incompresibles (ver, i.e., [23]), y buscamos un punto silla sobre el espacio $\mathbf{V}_N \times L_2(\Omega)$ (con $\mathbf{V}_N = \{ \mathbf{v} \in H(\text{div}; \Omega) : \mathbf{v} \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_N \}$) de un funcional Lagrangiano L definido por

$$L(\mathbf{v}, \mu) = J(\mathbf{v}) + \int_{\Omega} \mu \nabla \cdot \mathbf{u} \, d\mathbf{x}. \quad (1.6)$$

Si el par (\mathbf{u}, λ) es ese punto silla, entonces verifica (por definición)

$$\begin{cases} (\mathbf{u}, \lambda) \in \mathbf{V}_N \times L_2(\Omega), \\ L(\mathbf{u}, \mu) \leq L(\mathbf{u}, \lambda) \leq L(\mathbf{v}, \lambda), \quad \forall (\mathbf{v}, \mu) \in \mathbf{V}_N \times L_2(\Omega), \end{cases}$$

lo que implica la siguiente caracterización:

$$\begin{cases} (\mathbf{u}, \lambda) \in \mathbf{V}_N \times L^2(\Omega), \\ \int_{\Omega} S(\mathbf{u} - \mathbf{u}^I) \cdot \mathbf{v} \, d\mathbf{x} + \int_{\Omega} \lambda \nabla \cdot \mathbf{v} \, d\mathbf{x} = 0, \quad \forall \mathbf{v} \in \mathbf{V}_N, \\ \nabla \cdot \mathbf{u} = 0. \end{cases} \quad (1.7)$$

Se sigue de (1.7) que (\mathbf{u}, λ) satisface (en el sentido de distribuciones):

$$\begin{cases} \mathbf{u} = \mathbf{u}^I + S^{-1} \nabla \lambda \text{ in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 \text{ in } \Omega, \\ \mathbf{u} \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_N, \\ \lambda = 0 \text{ on } \Gamma_D. \end{cases} \quad (1.8)$$

Las relaciones (1.8) implican que λ es la única solución del siguiente problema elíptico (del tipo Neumann-Dirichlet):

$$\begin{cases} -\nabla \cdot S^{-1} \nabla \lambda = \nabla \cdot \mathbf{u}^I \text{ in } \Omega, \\ \lambda = 0 \text{ on } \Gamma_D, \\ (S^{-1} \nabla \lambda + \mathbf{u}^I) \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_N, \end{cases} \quad (1.9)$$

En [24] se utilizaron dos enfoques numéricos para estudiar cómo las condiciones de frontera para λ pueden afectar la reconstrucción del campo vectorial. El primero consistió en resolver con el método de los elementos finitos el problema elíptico (1.9), y al algoritmo resultante se le llamó *E-algorithm*. Cuando Ω es un dominio rectangular, Γ_D se tomó como la parte superior, y en la parte restante (vertical) se impuso la condición de frontera $S^{-1} \nabla \lambda \cdot \hat{\mathbf{n}} = 0$, la cual es equivalente a suponer que $\mathbf{u} \cdot \hat{\mathbf{n}} = \mathbf{u}^I \cdot \hat{\mathbf{n}}$ en dichas paredes; al algoritmo resultante se le llamó *E2-algorithm*. El segundo enfoque estuvo basado en la formulación de punto silla del problema de optimización al cual uno aplica un algoritmo de gradiente conjugado (*CG-algorithm*), inspirado en una metodología exitosa en dinámica de fluidos computacional [23]. Este nuevo enfoque requiere condiciones de frontera para el campo vectorial pero no requiere condiciones de frontera explícitas para el multiplicador y produce mejores resultados. En el presente trabajo introducimos un algoritmo de gradiente conjugado preconditionado asociado a este segundo enfoque, reduciendo el costo computacional. También mostramos vía experimentos numéricos que es aceptable simplificar el modelo tomando $S = I$ (la matriz identidad), al menos para los ejemplos numéricos sintéticos en 2-D considerados en esta tesis.

1.2 Control y estabilización de un Arreglo de Memoria de Juntas de Josephson (AMJJ)

El efecto Josephson es el fenómeno de supercorriente –i.e. una corriente que fluye indefinidamente sin aplicar ningún voltaje– que se obtiene a través de un dispositivo conocido como junta de

Josephson (JJ), la cual consiste de dos superconductores acoplados por un conector débil. El conector débil puede ser una barrera aislante delgada (conocida como una junta superconductor–aislante–superconductor, o S-I-S por sus siglas en inglés), una sección delgada de un metal no-superconductor (S-N-S), o una constricción física que debilita la superconductividad en el punto de contacto (S-s-S).

El efecto Josephson es un ejemplo de un fenómeno cuántico macroscópico. Se le dió este nombre después de que el físico británico Brian David Josephson estableciera en 1992 la relación matemática entre la corriente y el voltaje a través del conector débil. Las ecuaciones básicas que gobiernan la dinámica del efecto Josephson son

$$\begin{aligned} V &= \frac{\hbar}{2e} \frac{d\phi}{dt}, \\ I &= I_c \sin(\phi), \end{aligned}$$

donde $V(t)$ e $I(t)$ son el voltaje y la corriente a través de la junta de Josephson, ϕ es la ‘diferencia de fase’ a través de la junta (i.e., el factor de la diferencia en fase, o equivalentemente, argumento, entre el parámetro de orden complejo de Ginzburg–Landau de los dos superconductores que forman la junta), e I_c es una constante, la corriente crítica de la junta. Esta corriente crítica es un parámetro fenomenológico importante del dispositivo que puede verse afectado tanto por la temperatura como por la aplicación de un campo magnético. La constante física $\frac{\hbar}{2e}$ es el flujo cuántico magnético, el inverso de la constante de Josephson.

El efecto Josephson ha tenido una amplia gama de aplicaciones y podría ser una alternativa para los dispositivos de memoria. En 2005, el reporte de La Agencia Nacional de Seguridad de Estados Unidos menciona en forma concluyente que los transistores estaban alcanzando rápidamente sus límites físicos y que el sucesor mas probable podría estar basado en las Juntas de Josephson [30]. “Los circuitos de flujo cuántico simple (SFQ en Inglés) producen pulsos de corriente pequeños que viajan a casi 1/3 la velocidad de la luz, c . Las líneas de transmisión pasiva de superconductividad (PTL por sus siglas en inglés) son también capaces de transmitir los pulsos con pérdidas extremadamente pequeñas” [31]. Estos son los circuitos digitales de ‘switch’ más rápidos. “Las juntas de Josephson, dispositivos de conmutación (‘switching’) en superconductividad, cambian rápidamente (~ 1 ps), disipan poca energía en cada cambio ($< 10^{-19} J$), y comunican información vía pulsos de corriente que se propagan sobre las líneas de transmisión superconductoras casi sin pérdida” [32].

Gracias a las investigaciones de Y. Braiman y B. Neschke [33], la modelación de arreglos de memoria con Juntas de Josephson (AMJJ) ha progresado significativamente, incluyendo la identificación de sus configuraciones de estado estable, permitiendo operaciones de *Lectura* y *Escritura* usando un circuito de tres juntas acopladas inductivamente, modelado por las ecuaciones adimensionales

$$\begin{cases} \frac{d^2\phi_1}{dt^2} + \gamma_1 \frac{d\phi_1}{dt} + \kappa_1(\phi_1 - \phi_2) + \sin \phi_1 = I_1, \\ \frac{d^2\phi_2}{dt^2} + \gamma_2 \frac{d\phi_2}{dt} + \kappa_1(\phi_2 - \phi_1) + \kappa_2(\phi_2 - \phi_3) + \sin \phi_2 = I_2, \\ \frac{d^2\phi_3}{dt^2} + \gamma_3 \frac{d\phi_3}{dt} + \kappa_2(\phi_3 - \phi_2) + \sin \phi_3 = I_3, \end{cases} \quad (1.10)$$

donde $I_j = i_j + ad_j$, i_j es corriente directa y ad_j es una energía adicional, como un pulso. Los valores típicos para las distintas cantidades en el modelo son:

$$\gamma_1 = .7, \gamma_2 = 1.1, \gamma_3 = .7, i_1 = 1, i_2 = .8, i_3 = -1, \quad (1.11)$$

$$\kappa_1 = \kappa_2 = 0.1. \quad (1.12)$$

Los estados de equilibrio $\{\theta_1, \theta_2, \theta_3\}$ son las soluciones de

$$\begin{cases} \kappa_1(\phi_1 - \phi_2) + \sin \phi_1 = i_1, \\ \kappa_1(\phi_2 - \phi_1) + \kappa_2(\phi_2 - \phi_3) + \sin \phi_2 = i_2, \\ \kappa_2(\phi_3 - \phi_2) + \sin \phi_3 = i_3, \end{cases} \quad (1.13)$$

Para aquellos regímenes donde las segundas derivadas se pueden despreciar, (1.10) se reduce a

$$\begin{cases} \gamma_1 \frac{d\phi_1}{dt} + \kappa_1(\phi_1 - \phi_2) + \sin \phi_1 = I_1, \\ \gamma_2 \frac{d\phi_2}{dt} + \kappa_1(\phi_2 - \phi_1) + \kappa_2(\phi_2 - \phi_3) + \sin \phi_2 = I_2, \\ \gamma_3 \frac{d\phi_3}{dt} + \kappa_2(\phi_3 - \phi_2) + \sin \phi_3 = I_3. \end{cases} \quad (1.14)$$

Las operaciones de *Lectura/Escritura* pueden llevarse a cabo manipulando un par de estados estables aplicando pulsos Gaussianos apropiados, pero en este trabajo investigamos un enfoque de controlabilidad de (1.14) para optimizar la transición de una configuración estable a otra. Nuestro enfoque está muy relacionado con la metodología desarrollada en [34]. Esto significa que deseamos resolver el siguiente problema (o sus generalizaciones al variar la junta donde actúa el control o al involucrar mas juntas sobre las cuales actúen controles):

$$\begin{cases} u \in \mathcal{U}_{ad}, \\ J(u) \leq J(v), \forall v \in \mathcal{U}_{ad}, \end{cases} \quad (1.15)$$

donde (con $k > 0$ y $\|\cdot\|$ la norma Euclideana canónica) tenemos

$$J(v) = \frac{1}{2} \int_0^T v^2 dt + \frac{k}{2} \|\mathbf{y}(T) - \mathbf{y}_T\|^2; \quad (1.16)$$

siendo la función vectorial $\mathbf{y} = \{y_1, y_2, y_3\}$ la solución del siguiente problema de valores iniciales:

$$\begin{cases} \gamma_1 \frac{dy_1}{dt} + \kappa_1(y_1 - y_2) + \sin y_1 = i_1 + v \text{ en } (0, T), \\ \gamma_2 \frac{dy_2}{dt} + \kappa_1(y_2 - y_1) + \kappa_2(y_2 - y_3) + \sin y_2 = i_2 \text{ en } (0, T), \\ \gamma_3 \frac{dy_3}{dt} + \kappa_2(y_3 - y_2) + \sin y_3 = i_3 \text{ en } (0, T). \\ \mathbf{y}(0) = \mathbf{y}_0. \end{cases} \quad (1.17)$$

En (1.16)–(1.17), \mathbf{y}_0 y \mathbf{y}_T son estados de equilibrio inicial y final, respectivamente, y k es un parámetro de penalización. En cuanto al espacio de control \mathcal{U}_{ad} , podemos decir que es un subespacio de $L^2(0, T)$. Además de lo anterior, trabajaremos sobre la estabilización del sistema (1.14) alrededor de un estado de equilibrio inestable vía un proceso de control.

El trabajo de tesis está organizado en la forma siguiente. En la sección 2.1 consideramos el modelo matemático para ajustar campos de velocidad el cual está basado en una formulación de mínimos cuadrados, pero con el funcional de costo en un espacio de Hilbert adecuado donde se garantiza una solución única. En la sección 2.2 describimos la solución del problema de ajuste

de campos de velocidad, siguiendo el enfoque de punto silla. En la sección 2.3, después de reformular el problema, introducimos un algoritmo de gradiente conjugado preconditionado (*PCG–algorithm*), donde se usa un método de elementos finitos mixtos para resolver los subproblemas elípticos en cada iteración. En la sección 2.5 presentamos y discutimos los resultados numéricos, y en la sección 2.6 damos algunas conclusiones puntualizadas asociadas con el primer problema que se resuelve en esta tesis.

En los capítulos 3 y 4 tratamos con dos aspectos del control sobre el modelo AMJJ. En el capítulo 3 nos concentramos en cómo optimizar, vía un enfoque de controlabilidad, la transición del sistema entre dos configuraciones de equilibrio específicas, permitiendo por ejemplo, operaciones de *Lectura/Escritura*. El capítulo 4 trata lo concerniente a la estabilización del sistema alrededor de un equilibrio inestable vía un proceso de controlabilidad.

Capítulo 2

Reconstrucción de campos vectoriales con preconditionamiento óptimo

2.1 Formulación matemática y el problema de punto silla

Sea Ω una región abierta, simplemente conexa y acotada en \mathbb{R}^d ($d = 2$ ó 3) con frontera Lipschitz $\Gamma = \Gamma_N \cup \Gamma_D$, donde Γ_N es la parte de la frontera donde conocemos (i.e. sabemos cómo establecer) condiciones de frontera para la velocidad, y $\Gamma_D = \Gamma \setminus \Gamma_N$. De hecho, para recuperación de campos de viento especificamos para la velocidad las condiciones de frontera $\mathbf{u} \cdot \hat{\mathbf{n}} = 0$ en la superficie del terreno y $\mathbf{u} \cdot \hat{\mathbf{n}} = \mathbf{u}^I \cdot \hat{\mathbf{n}}$ en las fronteras verticales truncadas. Entonces, Γ_N incluye las fronteras inferior y verticales y Γ_D es la frontera superior, como se indica en la figura 2.1.

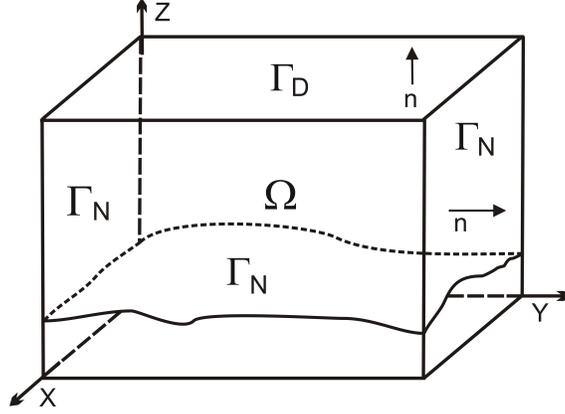


Figura 2.1: Dominio General.

En este caso, dado un campo vectorial inicial \mathbf{u}^I en Ω (el cual puede ser obtenido por interpolación de datos disponibles, o por otros medios), nuestro objetivo es generar un campo solenoidal \mathbf{u} –llamado campo ajustado– tan cercano a \mathbf{u}^I como sea posible en un sentido que será clarificado enseguida.

Tomando en cuenta el trabajo en [24], suponemos por el momento que $\mathbf{u} \cdot \hat{\mathbf{n}} = 0$ en Γ_N , y consideramos el espacio \mathbf{V} definido por (1.4) y equipado con la norma $\|\cdot\|_{S,\Omega}$ asociada con el producto interno

$$\langle \mathbf{u}, \mathbf{v} \rangle_S = \int_{\Omega} (S\mathbf{u}) \cdot \mathbf{v} \, d\mathbf{x},$$

donde $\mathbf{v} \cdot \mathbf{w} = \sum_{i=1}^d v_i w_i$ es el producto escalar usual en \mathbb{R}^d . Definiendo el funcional $J : \mathbf{V} \rightarrow \mathbb{R}$, por

$$J(\mathbf{v}) = \frac{1}{2} \|\mathbf{v} - \mathbf{u}^I\|_{S,\Omega}^2 = \frac{1}{2} \int_{\Omega} S(\mathbf{v} - \mathbf{u}^I) \cdot (\mathbf{v} - \mathbf{u}^I) \, d\mathbf{x}, \quad (2.1)$$

entonces el problema de generar el campo vectorial ajustado \mathbf{u} es el problema (1.3) ya planteado en la introducción, a saber

$$\text{Dado } \mathbf{u}^I \in \mathbf{H}(\text{div}; \Omega), \text{ encontrar } \mathbf{u} \in \mathbf{V} \text{ tal que } J(\mathbf{u}) \leq J(\mathbf{v}), \forall \mathbf{v} \in \mathbf{V}. \quad (2.2)$$

Las condiciones de primer y segundo orden para que J tenga un mínimo $\mathbf{u} \in \mathbf{V}$ son

$$\int_{\Omega} S(\mathbf{u} - \mathbf{u}^I) \cdot \mathbf{v} \, d\mathbf{x} = 0, \quad \forall \mathbf{v} \in \mathbf{V}, \quad (2.3)$$

$$\int_{\Omega} S\mathbf{v} \cdot \mathbf{v} \, d\mathbf{x} > 0, \quad \forall \mathbf{v} \in \mathbf{V}, \mathbf{v} \neq \mathbf{0}. \quad (2.4)$$

respectivamente. Dado que S es definida positiva, entonces (2.4) se da y la relación (2.3) se convierte en una condición necesaria y suficiente para que J tenga un mínimo global $\mathbf{u} \in V$. El teorema de Lax–Milgram garantiza que (2.3) tiene una única solución. Esta solución verifica las relaciones (1.8), las cuales implican que λ es la única solución en $H^1(\Omega)$ del problema elíptico (1.9) (un dual del problema (2.3)). No obstante que el problema (1.9) es bien planteado, el enfoque para obtener la solución de (2.3) no siempre es adecuado numéricamente, como se muestra en [24].

Como se mencionó en la introducción, una formulación de punto silla se obtiene a partir del Lagrangiano

$$L(\mathbf{v}, q) = J(\mathbf{v}) + \int_{\Omega} q \nabla \cdot \mathbf{v} \, d\mathbf{x}. \quad (2.5)$$

definido en $\mathbf{V}_N \times L_2(\Omega)$, donde \mathbf{V}_N es el espacio de funciones vectoriales dado por

$$\mathbf{V}_N = \{ \mathbf{v} \in H(\text{div}; \Omega) : \mathbf{v} \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_N \}. \quad (2.6)$$

Si un par $(\mathbf{u}, \lambda) \in \mathbf{V}_N \times L_2(\Omega)$ es un punto estacionario del Lagrangiano (2.5), entonces verifica

$$\int_{\Omega} S\mathbf{u} \cdot \mathbf{v} \, d\mathbf{x} + \int_{\Omega} \lambda \nabla \cdot \mathbf{v} \, d\mathbf{x} = \int_{\Omega} S\mathbf{u}^I \cdot \mathbf{v} \, d\mathbf{x}, \quad \forall \mathbf{v} \in \mathbf{V}_N, \quad (2.7)$$

$$\int_{\Omega} q \nabla \cdot \mathbf{u} \, d\mathbf{x} = 0, \quad \forall q \in L_2(\Omega). \quad (2.8)$$

Nota. El caso donde el campo vectorial recobrado \mathbf{u} tiene condiciones de frontera no homogéneas en Γ_N puede ser tratado en una forma similar separando \mathbf{u} en Γ_N y en el interior del dominio.

2.2 Un operador para el multiplicador de Lagrange

Hay técnicas numéricas efectivas para resolver problemas de punto silla como el sistema variacional (2.7)–(2.8). Aquí reformulamos el problema como una ecuación operacional en $L_2(\Omega)$ que tiene como solución a λ . Asumamos que (\mathbf{u}, λ) es solución del problema (2.7)–(2.8) con

$$\mathbf{u} = \mathbf{u}^I + \mathbf{u}_{\lambda}, \quad (2.9)$$

donde la velocidad \mathbf{u}^I es tangente a Γ_N y donde \mathbf{u}_{λ} es solución del siguiente problema variacional lineal

$$\left\{ \begin{array}{l} \mathbf{u}_{\lambda} \in \mathbf{V}_N, \\ \int_{\Omega} (S\mathbf{u}_{\lambda}) \cdot \mathbf{v} \, d\mathbf{x} = - \int_{\Omega} \lambda \nabla \cdot \mathbf{v} \, d\mathbf{x}, \quad \forall \mathbf{v} \in \mathbf{V}_N. \end{array} \right. \quad (2.10)$$

Tomando en cuenta que $\nabla \cdot \mathbf{u} = 0$, se sigue de (2.9) que $-\nabla \cdot \mathbf{u}_{\lambda} = \nabla \cdot \mathbf{u}^I$; esta ecuación puede ser expresada en forma operacional como

$$A \lambda = \nabla \cdot \mathbf{u}^1, \quad (2.11)$$

donde $A : L_2(\Omega) \rightarrow L_2(\Omega)$ es el operador definido por

$$A \mu = -\nabla \cdot \mathbf{u}_\mu, \quad (2.12)$$

con \mathbf{u}_μ la solución de

$$\begin{cases} \mathbf{u}_\mu \in \mathbf{V}_N, \\ \int_{\Omega} (S \mathbf{u}_\mu) \cdot \mathbf{v} \, d\mathbf{x} = - \int_{\Omega} \mu \nabla \cdot \mathbf{v} \, d\mathbf{x}, \quad \forall \mathbf{v} \in \mathbf{V}_N. \end{cases} \quad (2.13)$$

Aprovechando las propiedades del operador A (lineal, autoadjunto y fuertemente elíptico), la ecuación (2.11) puede ser resuelta por un algoritmo de gradiente conjugado operando en \mathbf{V}_N (como se muestra por ejemplo en [[23], Capítulos 3 y 4] para problemas variacionales lineales generales con las mismas propiedades que (2.13). Este algoritmo se puede consultar en la primera parte del Apéndice A de esta tesis.

2.3 Algoritmo de gradiente conjugado y su preconditionamiento óptimo.

El *CG-algorithm* de [24] proporciona resultados numéricos excelentes, principalmente en la reducción del promedio de la divergencia discreta, lo cual es el principal objetivo cuando se necesita un campo vectorial solenoidal. Sin embargo, el número de iteraciones requeridas para la convergencia es dependiente de la malla y se pueden requerir varios cientos o incluso más de mil, en algunos casos. Afortunadamente, hemos podido derivar un excelente preconditionador (de hecho, óptimo), el cual está basado en las propiedades del operador A . La idea es descrita a continuación. Sea $B : L_2(\Omega) \rightarrow L_2(\Omega)$ el operador definido por

$$Bq = \phi_q, \quad (2.14)$$

donde ϕ_q resuelve el problema

$$\int_{\Omega} (S^{-1} \nabla \phi_q) \cdot \nabla \psi \, d\mathbf{x} = \int_{\Omega} q \psi \, d\mathbf{x}, \quad \forall \psi \in H_D^1(\Omega), \quad (2.15)$$

$$\phi_q = 0 \quad \text{on } \Gamma_D, \quad (2.16)$$

$$S^{-1} \nabla \phi_q \cdot \hat{\mathbf{n}} = 0 \quad \text{on } \Gamma_N, \quad (2.17)$$

con

$$H_D^1(\Omega) = \{\psi \in H^1(\Omega) : \psi = 0 \text{ on } \Gamma_D\}. \quad (2.18)$$

El operador B satisface

$$\int_{\Omega} q' B q \, d\mathbf{x} = \int_{\Omega} q' \phi_q \, d\mathbf{x} = \int_{\Omega} S^{-1} \nabla \phi_{q'} \cdot \nabla \phi_q \, d\mathbf{x}, \quad \forall q, q' \in L_2(\Omega), \quad (2.19)$$

$$\int_{\Omega} q B q \, d\mathbf{x} = \int_{\Omega} S^{-1} \nabla \phi_q \cdot \nabla \phi_q \, d\mathbf{x} > c \|\nabla \phi_q\|_{L_2(\Omega)}^2, \quad \forall q \neq 0, \quad (2.20)$$

donde $0 < c < \min\{S_{ii}\}$, así que B es elíptico y autoadjunto (simétrico). Además también, satisface $ABq = q$ para cada $q \in L^2(\Omega)$. Esto se muestra formalmente enseguida, sin considerar condiciones de frontera:

$$Aq = -\nabla \cdot \mathbf{u}_q = -\nabla \cdot (S^{-1} \nabla q), \quad \text{ya que } S \mathbf{u}_q = \nabla q \text{ en } \Omega, \quad (2.21)$$

$$Bq = \phi_q = -[\nabla \cdot (S^{-1} \nabla)]^{-1} q, \quad \text{ya que } -\nabla \cdot (S^{-1} \nabla \phi_q) = q \text{ en } \Omega. \quad (2.22)$$

Entonces, de (2.21)–(2.22), se sigue

$$A(Bq) = A\phi_q = -\nabla \cdot (S^{-1} \nabla \phi_q) = q. \quad (2.23)$$

De hecho, podemos tomar B^{-1} como un *precondicionador* para A . La propiedad (2.23) nos dice que este preconditionador es óptimo, y que el algoritmo de gradiente conjugado preconditionado asociado (*PCG–algorithm*) convergerá en pocas iteraciones. No podemos esperar tener convergencia en una iteración debido a dos razones (al menos): tenemos que introducir condiciones de frontera para resolver el problema (2.21), las cuales están implícitas en la formulación débil (2.13); también, tenemos que resolver la versión discreta del problema. A continuación describimos el algoritmo *PCG–algorithm* en forma operacional para resolver (2.7)–(2.8):

1. **Inicialización:** λ^0 dado, $g^0 = A\lambda^0 - \nabla \cdot \mathbf{u}^I$, $\hat{g}^0 = Bg^0$, $d^0 = -\hat{g}^0$.
2. **Descenso:** Para $m \geq 0$, y asumiendo que conocemos λ^m , g^m , \hat{g}^m , d^m , encontrar λ^{m+1} , g^{m+1} , \hat{g}^{m+1} como

$$\begin{aligned} \lambda^{m+1} &= \lambda^m + \alpha_m d^m \quad \text{donde } \alpha_m = \langle g^m, \hat{g}^m \rangle / \langle d^m, A d^m \rangle. \\ g^{m+1} &= g^m + \alpha_m A d^m, \\ \hat{g}^{m+1} &= \hat{g}^m + \alpha_m B(A d^m). \end{aligned}$$

3. Prueba de convergencia y nueva dirección conjugada:

Si $\langle g^{m+1}, \hat{g}^{m+1} \rangle \leq \epsilon \langle g^0, \hat{g}^0 \rangle$, tomar $\lambda = \lambda^{m+1}$ y parar.

Si no $d^{m+1} = -\hat{g}^{m+1} + \beta_m d^m$ con $\beta_m = \frac{\langle g^{m+1}, \hat{g}^{m+1} \rangle}{\langle g^m, \hat{g}^m \rangle}$

Hacer $m = m + 1$ y regresar a 2.

Usando las ecuaciones (2.12)–(2.13), las cuales definen el operador A , y las ecuaciones (2.14)–(2.15) que definen el operador B , el algoritmo de gradiente conjugado detallado con preconditionamiento es como sigue:

Inicialización

1. Dado $\lambda^0 \in L_2(\Omega)$, resolver

$$\begin{cases} \mathbf{u}_\lambda^0 \in \mathbf{V}_N, \\ \int_{\Omega} (S \mathbf{u}_\lambda^0) \cdot \mathbf{v} \, d\mathbf{x} = - \int_{\Omega} \lambda^0 \nabla \cdot \mathbf{v} \, d\mathbf{x}, \quad \forall \mathbf{v} \in \mathbf{V}_N. \end{cases}$$

2. Sea $g^0 = \nabla \cdot \mathbf{u}^0$, donde $\mathbf{u}^0 = \mathbf{u}_\lambda^0 + \mathbf{u}^1$.

3. Resolver

$$\begin{cases} \phi^0 \in H_D^1(\Omega), \\ \int_{\Omega} (S^{-1} \nabla \phi^0) \cdot \nabla \psi \, d\mathbf{x} = \int_{\Omega} g^0 \psi \, d\mathbf{x}, \quad \forall \psi \in H_D^1(\Omega). \end{cases}$$

4. Sea $\hat{g}^0 = \phi^0$, $d^0 = \hat{g}^0$.

Descenso

Para $m \geq 0$, suponiendo que λ^m , g^m , \hat{g}^m , d^m , \mathbf{u}^m son conocidos, calcular λ^{m+1} , g^{m+1} , \hat{g}^{m+1} , d^{m+1} y \mathbf{u}^{m+1} , usando los pasos siguientes:

5. Resolver

$$\begin{cases} \bar{\mathbf{u}}^m \in \mathbf{V}_N, \\ \int_{\Omega} (S \bar{\mathbf{u}}^m) \cdot \mathbf{v} \, d\mathbf{x} = - \int_{\Omega} d^m \nabla \cdot \mathbf{v} \, d\mathbf{x}, \quad \forall \mathbf{v} \in \mathbf{V}_N. \end{cases}$$

6. Sea $\bar{g}^m = \nabla \cdot \bar{\mathbf{u}}^m$.

7. Resolver

$$\begin{cases} \bar{\phi}^m \in H_D^1(\Omega), \\ \int_{\Omega} (S^{-1} \nabla \bar{\phi}^m) \cdot \nabla \psi \, d\mathbf{x} = \int_{\Omega} \bar{g}^m \psi \, d\mathbf{x}, \quad \forall \psi \in H_D^1(\Omega). \end{cases}$$

8. Sea $\alpha_m = \int_{\Omega} g^m \hat{g}^m \, d\mathbf{x} / \int_{\Omega} \bar{g}^m d^m \, d\mathbf{x}$.

9. Tomar

$$\begin{aligned} \lambda^{m+1} &= \lambda^m - \alpha_m d^m, \\ \mathbf{u}^{m+1} &= \mathbf{u}^m - \alpha_m \bar{\mathbf{u}}^m, \\ g^{m+1} &= g^m - \alpha_m \bar{g}^m, \\ \hat{g}^{m+1} &= \hat{g}^m - \alpha_m \bar{\phi}^m. \end{aligned}$$

Prueba de convergencia y nueva dirección de descenso

Si $\int_{\Omega} g^{m+1} \hat{g}^{m+1} \, d\mathbf{x} / \int_{\Omega} g^0 \hat{g}^0 \, d\mathbf{x} < \varepsilon$, entonces hacer $\lambda = \lambda^{m+1}$, $\mathbf{u} = \mathbf{u}^{m+1}$ y parar.
En otro caso, hacer lo siguiente

10. Calcular $\beta_m = \int_{\Omega} g^{m+1} \hat{g}^{m+1} d\mathbf{x} / \int_{\Omega} g^m \hat{g}^m d\mathbf{x}$
11. Tomar $d^{m+1} = \hat{g}^{m+1} + \beta_m d^m$.
12. Hacer $m = m + 1$ y retornar a 5.

Nótese que en este algoritmo, \mathbf{u} y λ se calculan simultáneamente. Comparando este algoritmo con la versión no preconditionada, encontramos fácilmente que el trabajo adicional está relacionado con los pasos 3 y 7. Así que, el costo adicional de este algoritmo en cada iteración es la solución del problema elíptico en el paso 7. Sin embargo, este costo adicional es compensado con dos buenas propiedades: a) el preconditionamiento reduce dramáticamente el número de iteraciones; b) hay una reducción significativa de los grados de libertad en la versión discreta del problema elíptico en los pasos 3 y 7. Clarificaremos este último punto después de haber discretizado el algoritmo por medio del método de elementos finitos, lo cual hacemos en seguida.

2.4 Discretización por medio de un método de elementos finitos mixto

Para aproximar las funciones pertenecientes a los espacios \mathbf{V}_N y $L_2(\Omega)$, haremos uso de la aproximación de elementos finitos del tipo *Bercovier–Pironneau* [25] (ver también [[23], Capítulo 5]). Este es un método mixto estable donde las funciones vectoriales en \mathbf{V}_N , tales como \mathbf{u}_λ^0 , \mathbf{u}^m y $\bar{\mathbf{u}}^m$, son aproximadas por polinomios continuos lineales por pedazos en una triangulación fina \mathcal{T}_h de Ω . Las funciones escalares en $L_2(\Omega)$, tales como λ^m , g^m , \bar{g}^m , \hat{g}^m , d^m , son también aproximadas con polinomios continuos lineales por pedazos, pero esta vez en una triangulación el doble de gruesa, \mathcal{T}_{2h} de Ω . La triangulación fina \mathcal{T}_h es obtenida de la triangulación gruesa a través de una subdivisión regular de cada triángulo $T \in \mathcal{T}_{2h}$, como se muestra en la Figura 2.2.

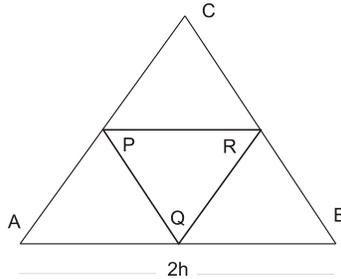


Figura 2.2: Elemento en \mathcal{T}_{2h} : triángulo ABC. Elementos en \mathcal{T}_h : triángulos AQP, PRC, PQR y QBR.

Entonces, los espacios de funciones \mathbf{V}_N y $L_2(\Omega)$ son aproximados por los siguientes subespacios de dimensión finita

$$\mathbf{V}_{Nh} = \{ \mathbf{v}_h \in C^0(\bar{\Omega})^2 : \mathbf{v}_h|_T \in P_1 \times P_1, \forall T \in \mathcal{T}_h, \mathbf{v}_h \cdot \mathbf{n} = 0 \text{ on } \Gamma_N \}, \quad (2.24)$$

y

$$L_{2h} = \{ q_h \in C^0(\bar{\Omega}) : q_h|_T \in P_1, \forall T \in \mathcal{T}_{2h} \}, \quad (2.25)$$

respectivamente. Aplicamos el método mixto descrito antes, particularmente en los pasos 1 y 5, así como en la versión débil de los pasos 2 y 6 del *PCG–algorithm*. En lo que respecta a los problemas elípticos en los pasos 3 y 7, éstos son aproximados sobre la triangulación gruesa \mathcal{T}_{2h} . Las funciones escalares en $H^1(\Omega)$, tales como ϕ^0 y $\bar{\phi}^m$ son aproximadas por polinomios lineales continuos por pedazos en cada uno de los triángulos de \mathcal{T}_{2h} . Entonces, $H^1(\Omega)$ es aproximado por medio del espacio de dimensión finita

$$H_{2h}^1 = \{ q_h \in C^0(\bar{\Omega}) : q_h|_T \in P_1, \forall T \in \mathcal{T}_{2h} \}, \quad (2.26)$$

Obsérvese que ya que \mathbf{u}_h es obtenida en la malla fina, su resolución es la misma que la obtenida con el algoritmo tradicional en \mathcal{T}_h . También, si la regla trapezoidal es aplicada para calcular las integrales del lado izquierdo en los pasos 1 y 5, obtenemos un sistema de ecuaciones algebraicas con matriz diagonal, y el costo de resolverlo es sólo una multiplicación de vectores. Entonces, el costo adicional del *PCG–algorithm* comparado con el costo del algoritmo *CG–Algorithm* es la solución de los problemas elípticos en los pasos 3 y 7, pero esos problemas son resueltos con una malla el doble de gruesa. Así que, para un problema bidimensional, el número de grados de libertad (número de incógnitas) en el problema algebraico resultante es cerca de cuatro veces menos que el número de grados de libertad obtenidos cuando se resuelven los problemas elípticos con el método tradicional descrito en la sección 1. Para un problema tridimensional el número de grados de libertad es cerca de ocho veces menos. De acuerdo a esto, el *PCG–algorithm* ahorra memoria en el almacenamiento matricial, cuando se compara con los algoritmos *E–algorithm* y *E2–algorithm* usados en [24]. La correspondiente matriz puede ser pre-calculada antes de iniciar las iteraciones ya que permanece constante a lo largo del proceso iterativo.

Para medir la diferencia global entre el campo exacto \mathbf{u} y el campo ajustado calculado \mathbf{u}_h tomamos el error relativo

$$e_r = \frac{\|\mathbf{u} - \mathbf{u}_h\|_2}{\|\mathbf{u}\|_2}, \quad (2.27)$$

y para medir qué tan cerca está \mathbf{u}_h de un campo vectorial solenoidal, calculamos la norma L^2 de la divergencia de \mathbf{u}_h , la cual denotamos por *ndiv*. En [24], la divergencia de \mathbf{u} en los vértices interiores \mathbf{x}_i fue aproximada en un sentido débil via la relación promediada

$$\nabla \cdot \mathbf{u}(\mathbf{x}_i) = -\frac{3}{|\Omega_i|} \int_{\Omega} \mathbf{u}_h \cdot \nabla \phi_i \, d\mathbf{x}, \quad (2.28)$$

donde en (2.28), ϕ_i es la función base lineal por pedazos asociada con \mathbf{x}_i , Ω_i es el interior del soporte $\bar{\Omega}_i$ de ϕ_i , y $|\Omega_i| = \text{medida}(\Omega_i)$. Como se mencionó antes, en el presente trabajo evaluamos directamente $\|\nabla \cdot \mathbf{u}_h\|_{L^2(\Omega)}$, en lugar de usar, como se hizo en [24], la suma pesada (por $\frac{|\Omega_i|}{3}$) de los valores en el lado derecho de (2.28) (cantidad denotada por *mdiv* en [24]).

2.5 Resultados numéricos

Para mostrar el desempeño del *PCG–algorithm* contra el *CG–algorithm* escogimos dos campos vectoriales solenoidales sintéticos. En el ejemplo 1 consideramos el campo vectorial 2-D $\mathbf{u}(x, y) = (x, -y)$ y en los otros ejemplos consideramos un flujo no viscoso alrededor de un cilindro circular. También consideramos dos casos: donde la componente vertical es recuperada de

datos horizontales (ejemplos 1, 2, 3) y donde el campo vectorial es recuperado de datos perturbados aleatoriamente (ejemplo 4). Todos los cálculos numéricos fueron hechos en una PC Toshiba: Portege R705, Windows 7, 64 Bits, Intel Processor CORE i3, 2.27 GHz, 3GB Ram.

En las tablas mostradas abajo, escribimos $S = I$ cuando la matriz diagonal S sea la matriz identidad. Cuando $S \neq I$, es suficiente dar los valores de S_{22} , ya que S_{11} se toma siempre como 1.

Ejemplo 1. Consideramos el campo vectorial solenoidal bidimensional $\mathbf{u}(x, y) = (x, -y)$ definido en $\Omega = (1, 2) \times (0, 1)$. Asumiendo que tenemos $\mathbf{u}^I(x, y) = (x, 0)$ como un campo vectorial horizontal inicial, deseamos ver que tanto podemos recobrar de la componente vertical de \mathbf{u} , aplicando los algoritmos *CG-algorithm* y *PCG-algorithm*. Un resumen de los resultados numéricos se encuentra en la tabla 2.1, donde mostramos el error relativo e_r , el promedio de la divergencia $mdiv$, así como el número de iteraciones para obtener convergencia, hasta una tolerancia dada ($\epsilon = 10^{-12}$, en los algoritmos *CG-algorithm* y *PCG-algorithm*) para diferentes tamaños de malla. Esta tabla también contiene el tiempo de CPU (seg) en cada experimento numérico, y, en particular, en la última columna se muestra el tiempo de CPU empleado por el *E2-algorithm* introducido en [24].

$S = I$	<i>CG-algorithm</i>				<i>PCG-algorithm</i>				<i>E2-alg</i>	
	Malla	e_r	$mdiv$	iters.	CPUt	e_r	$mdiv$	iters.	CPUt	CPUt
	33×33	3.1E-4	-7.6E-09	260	.25	1.9E-3	-3.4E-6	6	.03	.01
	65×65	1.1E-4	-5.0E-10	498	1.18	6.9E-4	-4.5E-7	7	.085	.06
	129×129	4.0E-5	-2.7E-11	854	7.24	2.4E-4	-5.9E-8	7	.26	.73
	257×257	1.4E-5	-1.6E-12	1688	60.69	8.6E-5	-7.5E-9	7	1.96	15.75

Tabla 2.1: Resultados numéricos para el ejemplo 1. $\epsilon = 10^{-12}$.

Es claro que el *PCG-algorithm* se desempeña mejor que los otros algoritmos. Otro buen aspecto del desempeño del *PCG-algorithm* en este ejemplo es que el número de iteraciones es independiente del tamaño de malla (7 iteraciones en cada caso). También, el error relativo entre la solución calculada y el campo vectorial exacto es del mismo orden para los algoritmos *CG-algorithm* y *PCG-algorithm*, en donde la diferencia mas grande ocurre en la frontera superior, ya que las condiciones de frontera no fueron impuestas explícitamente allí. En el resto de la frontera se impusieron condiciones de frontera exactas. Para este ejemplo, observamos una pérdida de precisión en la divergencia media cuando se utiliza el *PCG-algorithm*. Este fenómeno no ocurre con otros ejemplos más complicados y que se muestran adelante.. De cualquier manera, la divergencia media obtenida con el *PCG-algorithm* es todavía muy precisa, desde un punto de vista práctico, ya que la mayoría de los algoritmos numéricos tradicionales para forzar conservación de masa dan una divergencia promedio del orden de 10^{-2} como se muestra en la tabla 2.2, donde incluimos los resultados numéricos obtenidos en [24] con un tamaño de malla de $h = 1/80$ para la velocidad. En esos experimentos, el *E-algorithm* y el *E2-algorithm* corresponden a la solución del problema elíptico (1.9) con dos tipos diferentes de condiciones de frontera para λ .

En este ejemplo, el criterio usado para parar las iteraciones de gradiente conjugado, a saber, $\epsilon = 10^{-12}$, es muy estricto. Seleccionamos este valor sólomente porque deseabamos hacer comparaciones con los resultados numéricos obtenidos en [24]. De hecho, encontramos que el valor mucho menos restrictivo de $\epsilon = 10^{-4}$ aún proporciona muy buenos resultados, con menos esfuerzo computacional, por supuesto. La tabla 2.3 muestra los resultados numéricos obtenidos con este

Algoritmo	e_r	$mdiv$	iters.	CPUt
<i>E</i> -algorithm	1.9×10^{-2}	4.1×10^{-2}	—	1.78
<i>E2</i> -algorithm	4×10^{-4}	1.8×10^{-2}	—	1.78
<i>CG</i> -algorithm	5.9×10^{-4}	-5.3×10^{-12}	1214	3.9

Tabla 2.2: Resultados numéricos obtenidos en [24] para el ejemplo 1.

nuevo criterio de paro, para el mismo problema de prueba. Obsérvese que, en lugar de $mdiv$, incluimos esta vez la norma L_2 de la divergencia. Comparando con los resultados mostrados en la tabla 2.1, puede observarse que la pérdida de precisión (ya que son necesarias mucho menos iteraciones) no es muy significativa, especialmente cuando se utiliza el método de gradiente conjugado preconditionado. Es obvio que ésta es una forma mas eficiente para recuperar el mismo campo vectorial. Por esto, decidimos fijar la tolerancia ϵ en 10^{-4} para los siguientes ejemplos en este capítulo.

$S = I$	<i>CG</i> -algorithm			<i>PCG</i> -algorithm		
	e_r	$ndiv$	iters.	e_r	$ndiv$	iters.
Malla						
33×33	1.16E-3	1.24E-5	84	1.82E-3	6.19E-5	2
65×65	5.04E-4	3.29E-6	170	6.40E-4	1.09E-5	2
129×129	1.88E-4	9.04E-7	346	2.25E-4	1.93E-6	2
257×257	9.25E-5	2.23E-7	695	9.93E-5	3.40E-7	2

Tabla 2.3: Resultados numéricos para el ejemplo 1. $\epsilon = 10^{-4}$.

Ejemplo 2. En este caso consideramos un flujo no viscoso alrededor de un cilindro donde el campo vectorial, $\mathbf{u} = (u, w)$, se define por

$$u = U_0 + U_0 \frac{a^2}{r^4} (y^2 - x^2), \quad (2.29)$$

$$w = -2U_0 \frac{a^2}{r^4} xy, \quad (2.30)$$

$$r^2 = y^2 + x^2, \quad (2.31)$$

con $a = 1$, $U_0 = 0.01$. La figura 2.3 (izquierda) muestra este campo vectorial en el dominio $\Omega = (-2, 2) \times (0, 2) \setminus D$, donde D es la mitad superior del disco unitario.

De nuevo hacemos cero la componente vertical de la velocidad para obtener el campo vectorial inicial $\mathbf{u}^I = (u, 0)$. Las condiciones de frontera son las mismas que en el ejemplo previo, siendo la única diferencia la forma de la frontera inferior Γ_N donde el disco está localizado. Consideramos cuatro diferentes mallas para la velocidad, como se indica en la tabla 2.4, la primera ($h = 0.21$) se obtiene de una subdivisión regular de aquella mostrada en la figura 2.3 (derecha), y las otras mallas se obtuvieron por subdivisiones regulares adicionales. En las mallas subdivididas, los puntos medios entre dos nodos adyacentes en la frontera del disco son siempre trasladados a su punto mas cercano sobre la frontera del disco.

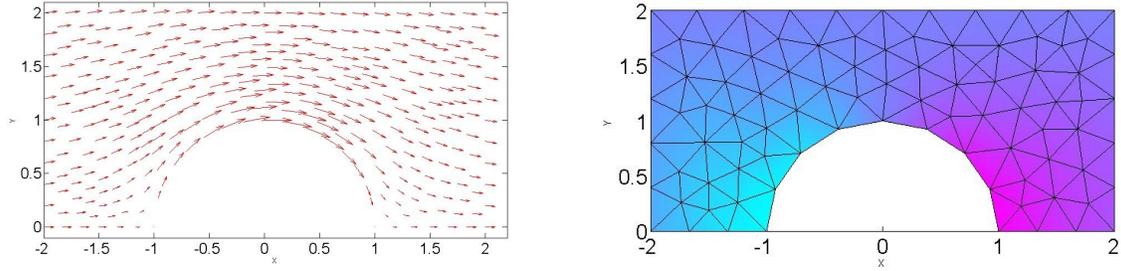


Figura 2.3: Izquierda: Campo vectorial. Derecha: Malla gruesa para el multiplicador en el ejemplo 2.

$S = I$	<i>CG-algorithm</i>			<i>PCG-algorithm</i>		
Malla	e_r	$ndiv$	iters.	e_r	$ndiv$	iters.
$h = 0.21$	8.91E-2	4.30E-5	17	8.93E-2	4.62E-5	5
$(1/2)h$	9.84E-2	1.17E-5	33	9.87E-2	1.54E-5	4
$(1/4)h$	1.04E-1	5.62E-6	68	1.04E-1	6.66E-6	4
$(1/8)h$	1.07E-1	3.59E-6	367	1.07E-1	4.86E-6	4

Tabla 2.4: Resultados numéricos para el Ejemplo 2.

La tabla 2.4 muestra que en todos los casos la norma L_2 de la divergencia es muy pequeña como ocurrió en el ejemplo 1. También, el número de iteraciones realizadas por el *CG-algorithm* para alcanzar la convergencia se incrementa al doble con cada refinamiento de la malla, mientras que el número de iteraciones con el *PCG-algorithm* es muy pequeño e independiente de la malla.

Por otro lado, los errores relativos e_r , que en este caso no son muy pequeños, se deben a que hay una inconsistencia entre el modelo invíscido y el problema mismo: es decir, si $S = I$ en (1.8) y λ es suave, entonces $\frac{\partial \lambda}{\partial x} = 0$ y $\frac{\partial \lambda}{\partial y} = w(x, y)$, lo que es posible si y solo si $w(x, y)$ no depende de x explícitamente. Esta condición no se satisface por el campo de velocidad (2.30), a diferencia de lo que sucede en el ejemplo 1, donde $w(x, y) = -y$. Por lo tanto, no podemos esperar un error relativo pequeño en este ejemplo; de hecho, la convergencia a la solución exacta se pierde. La figura 2.4 muestra la forma complicada del multiplicador λ_h .

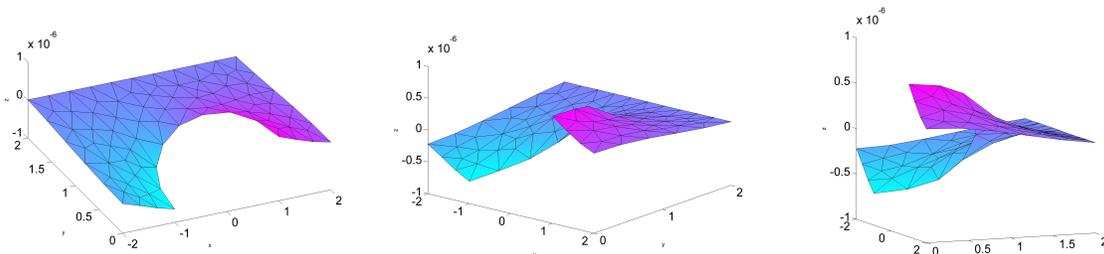


Figura 2.4: Vistas diferentes para el multiplicador λ en el ejemplo 2.

La figura 2.5, en la izquierda, muestra el correspondiente campo vectorial recobrado (en una

región cercana al disco); la misma figura, a la derecha, muestra la diferencia entre el campo exacto y el campo ajustado. Podemos ver que en este caso la distribución del error es compleja y la mayor diferencia ocurre cerca de la frontera del disco. Esas figuras dan una idea del nivel de dificultad para recobrar el campo vectorial en este ejemplo.

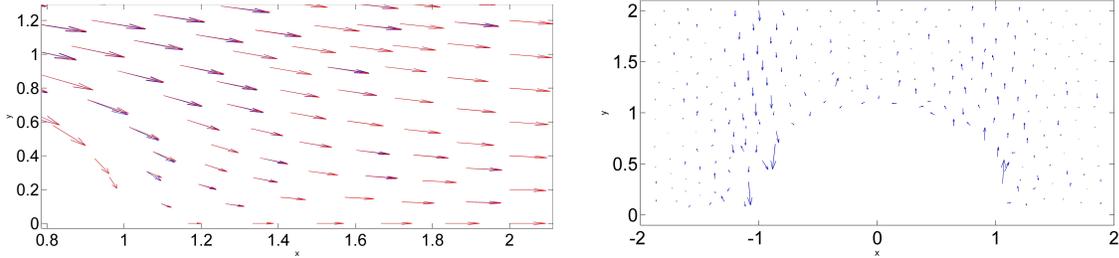


Figura 2.5: Izquierda: campos vectoriales ajustado (azul) y exacto (rojo) cerca del cilindro para el ejemplo 2. Derecha: diferencia entre los campos vectoriales exacto y ajustado (una ampliación).

Ejemplo 3. En este ejemplo consideramos los mismos campos vectoriales exacto e inicial como en el ejemplo 2, pero ahora en el dominio más simple $\Omega = (1, 5) \times (0, 2)$, donde el cilindro no está incluido. Consideramos cuatro mallas diferentes para la velocidad, como se indica en la tabla 2.5, la primera ($h = 0.2$) se obtiene con 20 subdivisiones horizontales y 10 subdivisiones verticales, mientras que las otras mallas se obtuvieron por subdivisiones regulares sucesivas de esta malla. La tabla 2.5 muestra que la norma L^2 de la divergencia es muy pequeña en todos los casos y se hace más pequeña cuando la malla se refina. Sin embargo, el error relativo no se reduce, pero no podemos esperar convergencia a la solución exacta debido a la inconsistencia explicada en el ejemplo previo.

$S = I$	<i>CG-algorithm</i>			<i>PCG-algorithm</i>			
	Malla	e_r	$ndiv$	iters.	e_r	$ndiv$	iters.
	20×10	5.84E-2	1.11E-5	23	5.86E-2	1.81E-5	4
	40×20	6.65E-2	1.88E-6	42	6.65E-2	3.48E-6	3
	80×40	7.07E-2	3.46E-7	81	7.07E-2	6.57E-7	2
	160×80	7.29E-2	7.08E-8	171	7.29E-2	1.26E-7	2
$S_{22} = 1E - 2$							
	20×10	8.78E-3	1.32E-5	42	8.96E-3	1.37E-5	8
	40×20	2.82E-3	1.23E-6	52	3.82E-3	2.55E-6	7
	80×40	4.18E-3	2.14E-7	103	4.35E-3	4.79E-7	4
	160×80	4.81E-3	4.55E-8	168	4.65E-3	9.44E-8	3

Tabla 2.5: Resultados numéricos para el ejemplo 3.

Para este ejemplo, hicimos varios experimentos con diferentes valores para S_{22} (recordar que $S_{11} = 1$), considerando potencias negativas y positivas de diez, i.e. $S_{22} = 10^n$ con $n \in \mathbb{Z}$. Encontramos que el mejor error relativo se dió con $S_{22} = 10^{-2}$. Valores más grandes que este

incrementan el error relativo y valores mas pequeños que este no mejoran significativamente los resultados. Por otro lado, la divergencia promedio es casi la misma que se obtuvo con $S = I$, pero a costa de mas iteraciones en el *PCG-algorithm*. Los mismos experimentos con diferentes valores para S_{22} también se realizaron para el problema del ejemplo 1. En ese caso el error relativo fué independiente de S en todos los casos, porque no hay inconsistencia entre el modelo, el campo vectorial de prueba y el campo vectorial inicial (esto es, $\frac{\partial \lambda}{\partial y}$ depende sólomente de y).

Ejemplo 4. En este ejemplo consideramos el mismo campo vectorial y el mismo dominio que en el ejemplo 3, pero ahora el campo inicial es $\mathbf{u}^I = (u + \delta u, w + \delta w)$ donde δu y δw son perturbaciones aleatorias de u y w de magnitud $|\delta u| \leq 0.3 |u|$ y $|\delta w| \leq 0.3 |w|$, respectivamente, en cada nodo de la malla. Esta vez aplicamos condiciones de frontera exactas para la velocidad y consideramos las mismas mallas que en el ejemplo 3. La tabla 2.6 muestra los resultados numéricos para las diferentes mallas.

$S = I$	<i>CG-algorithm</i>			<i>PCG-algorithm</i>		
Malla	e_r	$ndiv$	iters	e_r	$ndiv$	iters.
20×10	5.85E-2	1.94E-4	11	5.89E-2	1.95E-4	6
40×20	5.25E-2	5.64E-5	19	5.25E-2	5.66E-5	7
80×40	5.37E-2	2.47E-5	12	5.38E-2	2.48E-5	7
160×80	5.58E-2	1.16E-5	17	5.57E-2	1.17E-5	8

Tabla 2.6: Resultados numéricos para el Ejemplo 4.

Una vez más, la norma de la divergencia es pequeña en todos los casos, pero no decrece al mismo ritmo que en los ejemplos previos. Sin embargo, en este caso el error relativo es similar al obtenido en los ejemplos 2 y 3; pensamos que este comportamiento está asociado a la ya comentada inconsistencia del modelo invíscido, así como a la ausencia de convergencia debida a la perturbación aleatoria no suave. Las figuras 2.6 y 2.7 muestran la forma aleatoria del multiplicador λ , y el correspondiente campo vectorial recuperado.

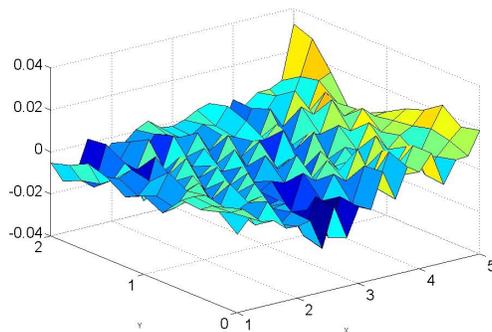


Figura 2.6: Multiplicador λ para el ejemplo 4.

La figura 2.8 muestra la diferencia entre el campo exacto y el campo ajustado, ilustrando que la distribución de los errores sigue un comportamiento aleatorio.

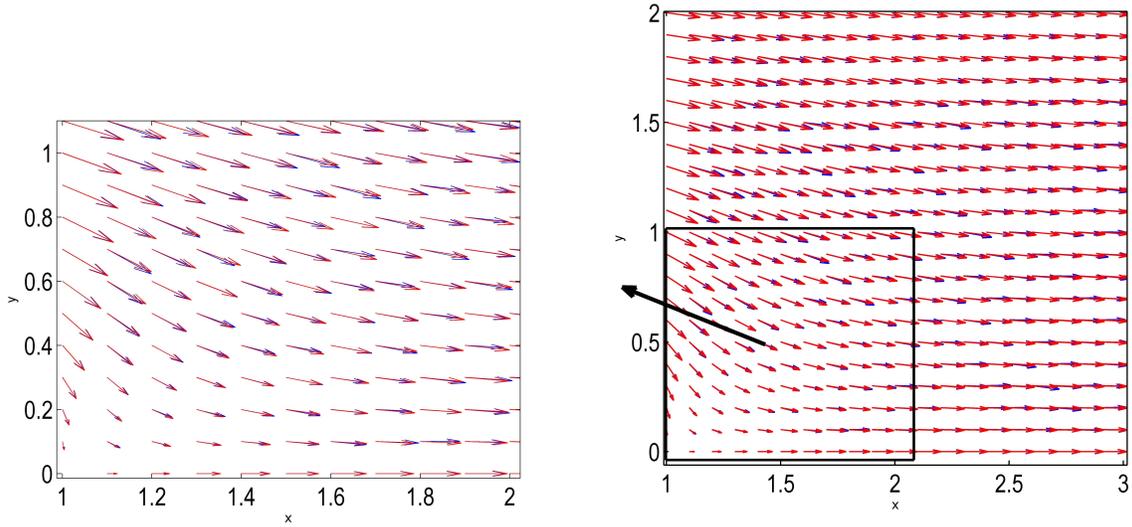


Figura 2.7: Campos vectoriales exacto (rojo) y ajustado (azul) para el ejemplo 4. A la izquierda se muestra una ampliación local.

Como en el ejemplo previo, hicimos varios experimentos con diferentes valores de S_{22} ($S_{11} = 1$). El mejor error relativo fue encontrado con $S_{22} = 5$. Con valores mas pequeños de S_{22} se incrementa el error relativo y con valores mas grandes de S_{22} los resultados no se mejoran significativamente. Otra vez, la divergencia promedio fué casi la misma que la obtenida con $S = I$, pero a un costo de mas iteraciones.

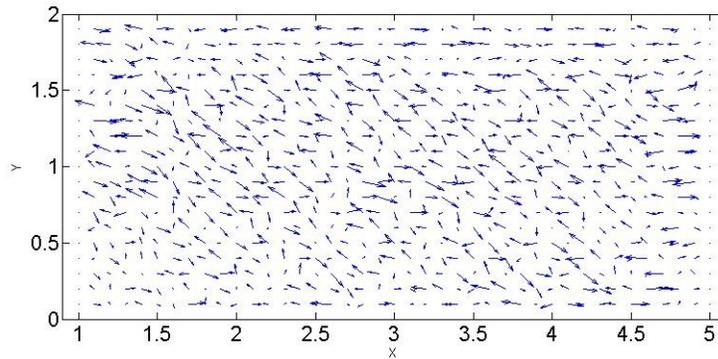


Figura 2.8: Diferencia entre el campo exacto y el campo recobrado en el ejemplo 4 (una ampliación).

2.6 Conclusiones

Siguiendo el estudio del modelo inviscido de Sasaki para recobrar campos vectoriales, hemos introducido un preconditionador óptimo para el algoritmo iterativo de gradiente conjugado, para resolver la ecuación operacional asociada con la formulación de punto silla del problema. Los resultados numéricos muestran que el nuevo algoritmo de gradiente conjugado preconditionado, y su aproximación estable por medio de una discretización de elementos finitos mixtos, preserva las buenas propiedades del algoritmo de gradiente conjugado no preconditionado, pero es mucho más rápido, y produce mejores resultados que los métodos tradicionales basados en el problema elíptico (1.8). Algunas buenas propiedades adicionales de este algoritmo son:

- Impone con mucha precisión la conservación de masa en el campo vectorial calculado, considerando que la aproximación es de segundo orden. También, encontramos que con $S = I$ podemos recuperar el campo vectorial con mucha precisión para todos los ejemplos. Sin embargo, cuando el modelo inviscido es inconsistente con el campo vectorial (i.e. $w(x, y)$ depende de x explícitamente), los resultados dependen de S . Muchos autores dan recomendaciones sobre cómo escoger S , [1, 5, 8, 11, 26, 27], pero no toman en cuenta que el modelo inviscido es inconsistente para recuperar ciertos campos vectoriales.
- El número de iteraciones es reducido de varios cientos a menos de diez para todos los ejemplos considerados en este trabajo, aún para dominios no triviales y mallas no uniformes. También, el número de iteraciones del *PCG-algorithm* es casi independiente con respecto al refinamiento de la malla, mientras que el número de iteraciones en el algoritmo no preconditionado, en muchos casos, aumenta al doble en cada refinamiento de la malla. De hecho, hay una reducción sustancial en el tiempo de cómputo en todos los casos. Sin embargo, este comportamiento debe ser corroborado en problemas 3-D con mallas adaptivas en dominios complejos, y con millones de grados de libertad, para tener una mejor idea del desempeño del método con escenarios realistas.
- No es necesario imponer condiciones de frontera al multiplicador, como se hace en los enfoques tradicionales. Mas aún, no se requiere un postproceso para encontrar el campo de velocidad a partir del multiplicador, dado que el multiplicador y el campo vectorial recuperado se encuentran simultáneamente dentro del algoritmo.

Por otro lado, hemos encontrado que el modelo inviscido puede tener limitaciones en algunos casos. Por ejemplo, cuando el campo vectorial inicial \mathbf{u}^1 es suave, el modelo inviscido puede ser inconsistente con el campo vectorial que queremos recuperar.

Esto se muestra en los ejemplos 2 y 3, donde el campo recuperado no converge al exacto conforme se refina la malla (ver el error relativo en las tablas 2.4 y 2.5). Es posible reducir el error relativo escogiendo una matriz diagonal S diferente de la identidad como se muestra en la tabla 2.5. Sin embargo, esta reducción no es significativa y depende de cada caso particular. En el ejemplo 1 (ver tabla 2.1) obtuvimos errores relativos pequeños porque no hay inconsistencia entre el modelo y el problema, mientras que en el ejemplo 4 no obtuvimos convergencia a la solución exacta, similar a los ejemplos 2 y 3.

La aplicación de la metodología presentada aquí a casos tridimensionales más realistas es un tema interesante para trabajo futuro. Otra cuestión interesante es la aplicación potencial de esas metodologías a otros campos experimentales tales como dinámica de fluidos y recuperación de

imágenes. En particular, recientemente nos hemos enterado acerca de la reconstrucción de campos vectoriales solenoidales a partir de datos experimentales obtenidos por medio de la técnica PIV (por sus siglas en inglés: Particle Image Velocimetry) [28] y aplicaciones potenciales a flujo óptimo [29]. A mediano plazo, esperamos establecer un modelo viscoso donde forcemos conservación de momento y conservación de energía, adicional a la conservación de masa.

Capítulo 3

Operadores de *Lectura/Escritura* para el AMJJ: Un enfoque de controlabilidad

3.1 Modelación matemática del AMJJ

Como mencionamos en la introducción, las ecuaciones adimensionales (t es también adimensional) para tres juntas de Josephson acopladas inductivamente son:

$$\begin{cases} \frac{d^2\phi_1}{dt^2} + \gamma_1 \frac{d\phi_1}{dt} + \kappa_1(\phi_1 - \phi_2) + \sin \phi_1 = I_1, \\ \frac{d^2\phi_2}{dt^2} + \gamma_2 \frac{d\phi_2}{dt} + \kappa_1(\phi_2 - \phi_1) + \kappa_2(\phi_2 - \phi_3) + \sin \phi_2 = I_2, \\ \frac{d^2\phi_3}{dt^2} + \gamma_3 \frac{d\phi_3}{dt} + \kappa_2(\phi_3 - \phi_2) + \sin \phi_3 = I_3. \end{cases} \quad (3.1)$$

Los estados de equilibrio $\{\theta_1, \theta_2, \theta_3\}$ son soluciones de

$$\begin{cases} \kappa_1(\phi_1 - \phi_2) + \sin \phi_1 = i_1, \\ \kappa_1(\phi_2 - \phi_1) + \kappa_2(\phi_2 - \phi_3) + \sin \phi_2 = i_2, \\ \kappa_2(\phi_3 - \phi_2) + \sin \phi_3 = i_3, \end{cases} \quad (3.2)$$

donde i_1, i_2 , e i_3 son **corrientes directas** e $I_j = i_j + ad_j$, con ad_j una energía adicional.

Para aquellos regímenes donde las segundas derivadas pueden ser despreciadas, (3.1) se reduce a

$$\begin{cases} \gamma_1 \frac{d\phi_1}{dt} + \kappa_1(\phi_1 - \phi_2) + \sin \phi_1 = I_1, \\ \gamma_2 \frac{d\phi_2}{dt} + \kappa_1(\phi_2 - \phi_1) + \kappa_2(\phi_2 - \phi_3) + \sin \phi_2 = I_2, \\ \gamma_3 \frac{d\phi_3}{dt} + \kappa_2(\phi_3 - \phi_2) + \sin \phi_3 = I_3. \end{cases} \quad (3.3)$$

Comenzaremos nuestras investigaciones sobre controlabilidad suponiendo que (3.3) es un modelo aceptable. De hecho, usar (3.1) en lugar de (3.3) no introduce dificultades conceptuales. Los valores típicos para las diferentes cantidades en el modelo son:

$$\gamma_1 = 0.7, \gamma_2 = 1.1, \gamma_3 = 0.7, i_1 = 1, i_2 = 0.8, i_3 = -1, \quad (3.4)$$

$$\kappa_1 = \kappa_2 = 0.1. \quad (3.5)$$

3.2 El problema de minimización

Cambiando a la notación que se usa comúnmente en teoría de control, formulamos como sigue el problema de minimización asociado con la búsqueda de una transición óptima desde una configuración de estado estacionario a otra:

$$\begin{cases} u \in \mathcal{U}_{ad}, \\ J(u) \leq J(v), \forall v \in \mathcal{U}_{ad}, \end{cases} \quad (3.6)$$

donde (con $k > 0$ y $\|\cdot\|$ la norma Euclidiana canónica), tenemos que

$$J(v) = \frac{1}{2} \int_0^T v^2 dt + \frac{k}{2} \|\mathbf{y}(T) - \mathbf{y}_T\|^2, \quad (3.7)$$

y la función vectorial $\mathbf{y} = \{y_1, y_2, y_3\}$ es la solución del siguiente problema de valores iniciales:

$$\begin{cases} \gamma_1 \frac{dy_1}{dt} + \kappa_1(y_1 - y_2) + \sin y_1 = i_1 + v \text{ en } (0, T), \\ \gamma_2 \frac{dy_2}{dt} + \kappa_1(y_2 - y_1) + \kappa_2(y_2 - y_3) + \sin y_2 = i_2 \text{ en } (0, T), \\ \gamma_3 \frac{dy_3}{dt} + \kappa_2(y_3 - y_2) + \sin y_3 = i_3 \text{ en } (0, T). \\ \mathbf{y}(0) = \mathbf{y}_0. \end{cases} \quad (3.8)$$

En (3.7), (3.8), \mathbf{y}_T es la función objetivo y k es un parámetro de penalización. Por el momento, no seremos muy específicos en cuanto al espacio de control \mathcal{U}_{ad} ; podemos decir que es un subconjunto del espacio $L^2(0, T)$.

3.3 Calculando $DJ(v)$

Es de interés teórico y práctico calcular el diferencial $DJ(v)$ de la función costo J en v . Para lograrlo usaremos un método de perturbaciones (cálculo variacional). Consideremos entonces una perturbación δv de la variable de control v . Tenemos así, con notación obvia,

$$\left\{ \begin{array}{l} \delta J(v) = \int_0^T DJ(v) \delta v dt \\ = \int_0^T v \delta v dt + k(\mathbf{y}(T) - \mathbf{y}_T) \cdot \delta \mathbf{y}(T), \end{array} \right. \quad (3.9)$$

donde en (3.9):

(i) Denotamos por $\mathbf{a} \cdot \mathbf{b}$ el producto punto de dos vectores \mathbf{a} y \mathbf{b} .

(ii) La función $\delta \mathbf{y} = \{\delta y_1, \delta y_2, \delta y_3\}$ es la solución del siguiente problema de valor inicial, obtenido al perturbar (3.8):

$$\left\{ \begin{array}{l} \gamma_1 \frac{d\delta y_1}{dt} + \kappa_1(\delta y_1 - \delta y_2) + \delta y_1 \cos y_1 = \delta v \text{ en } (0, T), \\ \gamma_2 \frac{d\delta y_2}{dt} + \kappa_1(\delta y_2 - \delta y_1) + \kappa_2(\delta y_2 - \delta y_3) + \delta y_2 \cos y_2 = 0 \text{ en } (0, T), \\ \gamma_3 \frac{d\delta y_3}{dt} + \kappa_2(\delta y_3 - \delta y_2) + \delta y_3 \cos y_3 = 0 \text{ en } (0, T). \\ \delta \mathbf{y}(0) = \mathbf{0}. \end{array} \right. \quad (3.10)$$

En forma matricial (3.10) queda como sigue:

$$\left\{ \begin{array}{l} \Gamma \frac{d}{dt} \delta \mathbf{y} + K \delta \mathbf{y} + \begin{pmatrix} \cos y_1 & 0 & 0 \\ 0 & \cos y_2 & 0 \\ 0 & 0 & \cos y_3 \end{pmatrix} \delta \mathbf{y} = \begin{pmatrix} \delta v \\ 0 \\ 0 \end{pmatrix} \text{ en } (0, T), \\ \delta \mathbf{y}(0) = \mathbf{0}. \end{array} \right. \quad (3.11)$$

con

$$\Gamma = \begin{pmatrix} \gamma_1 & 0 & 0 \\ 0 & \gamma_2 & 0 \\ 0 & 0 & \gamma_3 \end{pmatrix} \text{ y } K = \begin{pmatrix} \kappa_1 & -\kappa_1 & 0 \\ -\kappa_1 & \kappa_1 + \kappa_2 & -\kappa_2 \\ 0 & -\kappa_2 & \kappa_2 \end{pmatrix}.$$

Introducimos ahora una función vectorial $\mathbf{p} = \{p_1, p_2, p_3\}$ la cual suponemos diferenciable sobre $(0, T)$; multiplicando por \mathbf{p} ambos lados de la ecuación diferencial en (3.11), e integrando sobre $(0, T)$ obtenemos:

$$\begin{aligned} & \int_0^T \Gamma \frac{d}{dt} \delta \mathbf{y} \cdot \mathbf{p} dt + \int_0^T K \delta \mathbf{y} \cdot \mathbf{p} dt + \int_0^T \begin{pmatrix} \cos y_1 & 0 & 0 \\ 0 & \cos y_2 & 0 \\ 0 & 0 & \cos y_3 \end{pmatrix} \delta \mathbf{y} \cdot \mathbf{p} dt \\ & = \int_0^T p_1 \delta v dt \text{ en } (0, T). \end{aligned} \quad (3.12)$$

Integrando por partes, y tomando en cuenta la simetría de las matrices Γ y K , se sigue de (3.11), (3.12) que

$$\begin{aligned} & \Gamma \mathbf{p}(T) \cdot \delta \mathbf{y}(T) + \int_0^T -\Gamma \frac{d}{dt} \mathbf{p} \cdot \delta \mathbf{y} dt + \int_0^T K \mathbf{p} \cdot \delta \mathbf{y} dt \\ & + \int_0^T \begin{pmatrix} \cos y_1 & 0 & 0 \\ 0 & \cos y_2 & 0 \\ 0 & 0 & \cos y_3 \end{pmatrix} \mathbf{p} \cdot \delta \mathbf{y} dt \\ & = \int_0^T p_1 \delta v dt \text{ en } (0, T); \end{aligned}$$

así que

$$\begin{aligned} & \Gamma \mathbf{p}(T) \cdot \delta \mathbf{y}(T) + \int_0^T \left\{ -\Gamma \frac{d}{dt} \mathbf{p} + K \mathbf{p} + C(\mathbf{y}) \mathbf{p} \right\} \cdot \delta \mathbf{y} dt = \\ & = \int_0^T p_1 \delta v dt \text{ en } (0, T), \end{aligned} \tag{3.13}$$

con $C(\mathbf{y}) = \begin{pmatrix} \cos y_1 & 0 & 0 \\ 0 & \cos y_2 & 0 \\ 0 & 0 & \cos y_3 \end{pmatrix}$. Supóngase que la función \mathbf{p} es solución del sistema (**adjunto**) siguiente:

$$\begin{cases} -\Gamma \frac{d\mathbf{p}}{dt} + K \mathbf{p} + \begin{pmatrix} \cos y_1 & 0 & 0 \\ 0 & \cos y_2 & 0 \\ 0 & 0 & \cos y_3 \end{pmatrix} \mathbf{p} = \mathbf{0}, \\ \Gamma \mathbf{p}(T) = k(\mathbf{y}(T) - \mathbf{y}_T). \end{cases} \tag{3.14}$$

De (3.13), (3.14) se obtiene

$$k(\mathbf{y}(T) - \mathbf{y}_T) \cdot \delta \mathbf{y}(T) = \int_0^T p_1 \delta v dt. \tag{3.15}$$

Combinando (3.9) y (3.15) obtenemos

$$\int_0^T DJ(v) \delta v dt = \int_0^T v \delta v dt + k(\mathbf{y}(T) - \mathbf{y}_T) \cdot \delta \mathbf{y}(T) = \int_0^T (v + p_1) \delta v dt, \tag{3.16}$$

lo cual implica que

$$DJ(v) = v + p_1. \tag{3.17}$$

Nota 4.1. Hasta ahora, hemos supuesto que el control sólo actúa en la primera junta. Sin embargo, el método que hemos usado para calcular DJ también se puede aplicar cuando uno considera controlar la transición de \mathbf{y}_0 a \mathbf{y}_T usando un control actuando en otra junta, o usando dos o mas controles actuando en dos o mas juntas.

Nota 4.2. Si tomamos $\mathcal{U}_{ad} = L^2(0, T)$, es improbable que uno pueda contar con un dispositivo que lleve a cabo la acción del control óptimo asociado u ; sin embargo, el conocimiento de este control óptimo puede dar ideas acerca de cómo definir espacios de control mas prácticos. Esto justifica el hecho de que mas adelante discutamos el cálculo de u en (3.6) cuando $\mathcal{U}_{ad} = L^2(0, T)$, aprovechando la relación (3.17).

Nota 4.3. En lugar de (3.7) podemos usar el funcional

$$J(v) = \frac{\varepsilon}{2} \int_0^T v^2 dt + \frac{1}{2} \|\mathbf{y}(T) - \mathbf{y}_T\|^2. \quad (3.18)$$

3.4 Sobre la solución del problema de minimización (3.6) cuando $\mathcal{U}_{ad} = L^2(0, T)$

3.4.1 Sinopsis

Como el espacio $L^2(0, T)$ es un espacio de Hilbert con producto interior $\{v, w\} \rightarrow \int_0^T v w dt$, tiene sentido tratar de resolver el problema (3.6) usando uno de los algoritmos de gradiente conjugado discutidos en el capítulo 3 de [23] y resumidos en la segunda parte del Apéndice A.

En la subsección 3.4.2 aplicaremos el algoritmo mencionado en A.2 a la solución por gradiente conjugado del problema (3.6) cuando $\mathcal{U}_{ad} = L^2(0, T)$.

La implementación práctica del algoritmo de gradiente conjugado se discutirá en la sección 3.5.

3.4.2 Aplicación del algoritmo de gradiente conjugado (A.12)-(A.19) a la solución del problema (3.6)

Recordemos que $\mathcal{U}_{ad} = L^2(0, T)$ es un espacio de Hilbert para el producto interno $\{v, w\} \rightarrow \int_0^T v w dt$ y la norma asociada $v \rightarrow \sqrt{\int_0^T |v|^2 dt}$, por lo que el problema (3.6) es un caso particular del problema (A.9), así que podemos aplicar el algoritmo de gradiente conjugado (A.12)-(A.19) a la solución del problema (3.6). En nuestro caso particular, se sigue de (3.17) que ese algoritmo toma la forma siguiente:

- Suponer que

$$u^0 \text{ es dado en } L^2(0, T) \text{ (} u^0 = 0 \text{ por ejemplo)}. \quad (3.19)$$

- Resolver para \mathbf{y}^0

$$\left\{ \begin{array}{l} \Gamma \frac{d}{dt} \mathbf{y}^0 + K \mathbf{y}^0 + \begin{pmatrix} \sin y_1^0 \\ \sin y_2^0 \\ \sin y_3^0 \end{pmatrix} = \begin{pmatrix} i_1 + u^0 \\ i_2 \\ i_3 \end{pmatrix} \text{ en } (0, T), \\ \mathbf{y}^0(0) = \mathbf{y}_0. \end{array} \right. \quad (3.20)$$

y entonces para \mathbf{p}^0

$$\left\{ \begin{array}{l} -\Gamma \frac{d}{dt} \mathbf{p}^0 dt + K \mathbf{p}^0 dt + \begin{pmatrix} \cos y_1^0 & 0 & 0 \\ 0 & \cos y_2^0 & 0 \\ 0 & 0 & \cos y_3^0 \end{pmatrix} \mathbf{p}^0 = \mathbf{0}, \text{ en } (0, T), \\ \Gamma \mathbf{p}^0(T) = k(\mathbf{y}^0(T) - \mathbf{y}_T). \end{array} \right. \quad (3.21)$$

- Hacer

$$g^0 = u^0 + p_1^0. \quad (3.22)$$

- Si $\int_0^T |g^0|^2 dt \leq tol^2 \max[1, \int_0^T |u^0|^2 dt]$, tomar $u = u^0$; en otro caso, hacer

$$w^0 = g^0. \quad (3.23)$$

Para $q \geq 0$, u^q , g^q y w^q conocidos, los dos últimos diferentes de 0, calcular u^{q+1} y, si es necesario w^{q+1} como sigue:

- Resolver

$$\left\{ \begin{array}{l} \rho_q \in \mathbb{R}, \\ J(u^q - \rho_q w^q) \leq J(u^q - \rho w^q), \forall \rho \in \mathbb{R}. \end{array} \right. \quad (3.24)$$

- Hacer

$$u^{q+1} = u^q - \rho_q w^q. \quad (3.25)$$

- Resolver para \mathbf{y}^{q+1}

$$\left\{ \begin{array}{l} \Gamma \frac{d}{dt} \mathbf{y}^{q+1} + K \mathbf{y}^{q+1} + \begin{pmatrix} \sin y_1^{q+1} \\ \sin y_2^{q+1} \\ \sin y_3^{q+1} \end{pmatrix} = \begin{pmatrix} i_1 + u^{q+1} \\ i_2 \\ i_3 \end{pmatrix} \text{ en } (0, T), \\ \mathbf{y}^{q+1}(0) = \mathbf{y}_0. \end{array} \right. \quad (3.26)$$

y entonces para \mathbf{p}^{q+1}

$$\left\{ \begin{array}{l} -\Gamma \frac{d}{dt} \mathbf{p}^{q+1} dt + K \mathbf{p}^{q+1} dt \\ + \begin{pmatrix} \cos y_1^{q+1} & 0 & 0 \\ 0 & \cos y_2^{q+1} & 0 \\ 0 & 0 & \cos y_3^{q+1} \end{pmatrix} \mathbf{p}^{q+1} = \mathbf{0}, \text{ en } (0, T), \\ \Gamma \mathbf{p}^{q+1}(T) = k(\mathbf{y}^{q+1}(T) - \mathbf{y}_T). \end{array} \right. \quad (3.27)$$

- Hacer

$$g^{q+1} = u^{q+1} + p_1^{q+1}. \quad (3.28)$$

- Si

$$\frac{\int_0^T |g^{q+1}|^2 dt}{\max[\int_0^T |g^0|^2 dt, \int_0^T |u^{q+1}|^2 dt]} \leq tol^2,$$

tomar $u = u^{q+1}$; en otro caso, calcular

$$\beta_q = \frac{\int_0^T |g^{q+1}|^2 dt}{\int_0^T |g^q|^2 dt} \quad (3.29)$$

y hacer

$$w^{q+1} = g^{q+1} + \beta_q w^q. \quad (3.30)$$

- Hacer $q + 1 \rightarrow q$ y regresar a (3.24) .
- Fin del algoritmo.

La implementación práctica del algoritmo (3.19)-(3.30), vía una aproximación por diferencias finitas del problema (3.6), será discutida en la sección 3.5.

3.5 Sobre la implementación práctica del algoritmo (3.19)-(3.30) vía una aproximación por diferencias finitas del problema (3.6)

3.5.1 Aproximación por diferencias finitas del problema (3.6) cuando $\mathcal{U}_{ad} = L^2(0, T)$

Como se hace en [34] (para otros problemas), aproximamos (3.6) cuando $\mathcal{U}_{ad} = L^2(0, T)$ por

$$\begin{cases} \mathbf{u}^{\Delta t} \in \mathcal{U}_{ad}^{\Delta t}, \\ J^{\Delta t}(\mathbf{u}^{\Delta t}) \leq J^{\Delta t}(\mathbf{v}), \forall \mathbf{v} \in \mathcal{U}_{ad}^{\Delta t}, \end{cases} \quad (3.31)$$

donde:

- $\Delta t = T/N$ con N un entero positivo “grande” .
- $\mathcal{U}_{ad}^{\Delta t} = \mathbb{R}^N$.
- El funcional de costo $J^{\Delta t}$ se define por

$$J^{\Delta t}(\mathbf{v}) = \frac{\Delta t}{2} \sum_{n=1}^N |v^n|^2 + \frac{k}{2} \|\mathbf{y}^N - \mathbf{y}_T\|^2,$$

con $\mathbf{v} = \{v^n\}_{n=1}^N$ y $\{\mathbf{y}^n\}_{n=1}^N$ obteniéndose de \mathbf{v} y \mathbf{y}_0 via la siguiente variante discreta de (3.8):

$$\mathbf{y}^0 = \mathbf{y}_0, \quad (3.32)$$

y para $n = 1, \dots, N$

$$\left\{ \Gamma \frac{\mathbf{y}^n - \mathbf{y}^{n-1}}{\Delta t} + K\mathbf{y}^n + \begin{pmatrix} \sin y_1^{n-1} \\ \sin y_2^{n-1} \\ \sin y_3^{n-1} \end{pmatrix} = \begin{pmatrix} i_1 + v^n \\ i_2 \\ i_3 \end{pmatrix} \right. \text{ en } (0, T), \quad (3.33)$$

Para calcular \mathbf{y}^n necesitamos entonces resolver un sistema lineal del tipo siguiente:

$$(\Gamma + \Delta t K)\mathbf{y}^n = RHS^n. \quad (3.34)$$

La matriz

$$\begin{aligned} \Gamma + \Delta t K &= \begin{pmatrix} \gamma_1 & 0 & 0 \\ 0 & \gamma_2 & 0 \\ 0 & 0 & \gamma_3 \end{pmatrix} + \Delta t \begin{pmatrix} \kappa_1 & -\kappa_1 & 0 \\ -\kappa_1 & \kappa_1 + \kappa_2 & -\kappa_2 \\ 0 & -\kappa_2 & \kappa_2 \end{pmatrix} \\ &= \begin{pmatrix} \gamma_1 + \Delta t \kappa_1 & -\Delta t \kappa_1 & 0 \\ -\Delta t \kappa_1 & \gamma_2 + \Delta t(\kappa_1 + \kappa_2) & -\Delta t \kappa_2 \\ 0 & -\Delta t \kappa_2 & \gamma_3 + \Delta t \kappa_2 \end{pmatrix} \end{aligned}$$

al ser una matriz 3×3 simétrica y definida positiva, permite resolver (3.34) en forma particularmente fácil.

3.5.2 Cálculo de la derivada $DJ^{\Delta t}(v)$

Si uno desea usar el algoritmo de gradiente conjugado (A.12)-(A.19) para resolver el problema discreto (3.31), es necesario primero calcular $DJ^{\Delta t}(\mathbf{v})$. El método que usaremos para calcular $DJ^{\Delta t}(\mathbf{v})$ imita en mucho al método que usamos para calcular $DJ(\mathbf{v})$ en la sección 4.3. En $\mathcal{U}_{ad}^{\Delta t} = \mathbb{R}^N$ usaremos el producto interior siguiente

$$(\mathbf{v}, \mathbf{w})_{\Delta t} = \Delta t \sum_{n=1}^N v^n w^n, \quad \forall \mathbf{v} = \{v^n\}_{n=1}^N, \quad \mathbf{w} = \{w^n\}_{n=1}^N \in \mathbb{R}^N.$$

Tenemos

$$\delta J^{\Delta t}(\mathbf{v}) = \Delta t \sum_{n=1}^N v^n \delta v^n + k(\mathbf{y}^N - \mathbf{y}_T) \cdot \delta \mathbf{y}^N, \quad (3.35)$$

con $\{\delta \mathbf{y}^n\}_{n=0}^N$ obtenido por perturbación de (3.32), (3.33), esto es

$$\delta \mathbf{y}^0 = \mathbf{0}, \quad (3.36)$$

y para $n = 1, \dots, N$

$$\begin{aligned} &\Gamma \frac{\delta \mathbf{y}^n - \delta \mathbf{y}^{n-1}}{\Delta t} + K \delta \mathbf{y}^n + \begin{pmatrix} \cos y_1^{n-1} & 0 & 0 \\ 0 & \cos y_2^{n-1} & 0 \\ 0 & 0 & \cos y_3^{n-1} \end{pmatrix} \delta \mathbf{y}^{n-1} \\ &= \begin{pmatrix} \delta v^n \\ 0 \\ 0 \end{pmatrix}. \end{aligned} \quad (3.37)$$

Introducimos ahora $\{\mathbf{p}^n\}_{n=1}^N \in (\mathbb{R}^3)^N$. Tomando el producto punto de \mathbf{p}^n con cada lado de la ecuación (3.37), obtenemos, después de sumar y multiplicar por Δt :

$$\begin{aligned}
& \Delta t \sum_{n=1}^N \Gamma \frac{\delta \mathbf{y}^n - \delta \mathbf{y}^{n-1}}{\Delta t} \cdot \mathbf{p}^n + \Delta t \sum_{n=1}^N K \delta \mathbf{y}^n \cdot \mathbf{p}^n \\
& + \Delta t \sum_{n=1}^N \begin{pmatrix} \cos y_1^{n-1} & 0 & 0 \\ 0 & \cos y_2^{n-1} & 0 \\ 0 & 0 & \cos y_3^{n-1} \end{pmatrix} \delta \mathbf{y}^{n-1} \cdot \mathbf{p}^n \\
& = \Delta t \sum_{n=1}^N \delta v^n p_1^n.
\end{aligned} \tag{3.38}$$

Aplicando integración por partes discreta a la relación (3.38), y considerando que $\delta \mathbf{y}^0 = \mathbf{0}$, obtenemos:

$$\begin{aligned}
& \Gamma \mathbf{p}^{N+1} \cdot \delta \mathbf{y}^N + \Delta t \sum_{n=1}^N \Gamma \frac{\mathbf{p}^n - \mathbf{p}^{n+1}}{\Delta t} \cdot \delta \mathbf{y}^n + \Delta t \sum_{n=1}^N K \mathbf{p}^n \cdot \delta \mathbf{y}^n \\
& + \Delta t \sum_{n=1}^{N-1} \begin{pmatrix} \cos y_1^n & 0 & 0 \\ 0 & \cos y_2^n & 0 \\ 0 & 0 & \cos y_3^n \end{pmatrix} \mathbf{p}^{n+1} \cdot \delta \mathbf{y}^n \\
& = \Delta t \sum_{n=1}^N p_1^n \delta v^n.
\end{aligned} \tag{3.39}$$

o en forma equivalente

$$\begin{aligned}
& \Gamma \mathbf{p}^{N+1} \cdot \delta \mathbf{y}^N + \Delta t \left[\Gamma \frac{\mathbf{p}^N - \mathbf{p}^{N+1}}{\Delta t} + K \mathbf{p}^N \right] \cdot \delta \mathbf{y}^N \\
& + \Delta t \left[\sum_{n=1}^{N-1} \Gamma \frac{\mathbf{p}^n - \mathbf{p}^{n+1}}{\Delta t} + \sum_{n=1}^{N-1} K \mathbf{p}^n \right. \\
& \left. + \sum_{n=1}^{N-1} \begin{pmatrix} \cos y_1^n & 0 & 0 \\ 0 & \cos y_2^n & 0 \\ 0 & 0 & \cos y_3^n \end{pmatrix} \mathbf{p}^{n+1} \right] \cdot \delta \mathbf{y}^n \\
& = \Delta t \sum_{n=1}^N p_1^n \delta v^n.
\end{aligned} \tag{3.40}$$

Supóngase que $\{\mathbf{p}^n\}_{n=1}^{N+1}$ es solución del siguiente sistema adjunto discreto:

$$\Gamma \mathbf{p}^{N+1} = k(\mathbf{y}^N - \mathbf{y}_T), \tag{3.41}$$

$$\Gamma \frac{\mathbf{p}^N - \mathbf{p}^{N+1}}{\Delta t} + K \mathbf{p}^N = \mathbf{0}, \tag{3.42}$$

y para $n = N - 1, \dots, 1$

$$\Gamma \frac{\mathbf{p}^n - \mathbf{p}^{n+1}}{\Delta t} + K \mathbf{p}^n + \begin{pmatrix} \cos y_1^n & 0 & 0 \\ 0 & \cos y_2^n & 0 \\ 0 & 0 & \cos y_3^n \end{pmatrix} \mathbf{p}^{n+1} = \mathbf{0}. \tag{3.43}$$

Se sigue de (3.35), (3.40)-(3.43) que

$$\delta J^{\Delta t}(\mathbf{v}) = \Delta t \sum_{n=1}^N v^n \delta v^n + \Delta t \sum_{n=1}^N p_1^n \delta v^n = \Delta t \sum_{n=1}^N (v^n + p_1^n) \delta v^n, \quad (3.44)$$

esto es

$$DJ^{\Delta t}(\mathbf{v}) = \{v^n + p_1^n\}_{n=1}^N. \quad (3.45)$$

3.5.3 Solución por gradiente conjugado del problema de control discreto (3.31)

Para describir la variante discreta del algoritmo (3.19)-(3.30), la principal dificultad está asociada con la indexación; usaremos $\mathbf{y}_q^n = \{\mathbf{y}_{iq}^n\}_{i=1}^3$ para denotar el valor discreto de la función vectorial y al tiempo $n\Delta t$ en la iteración q ; similarmente, u_q^n denotará el valor discreto del control u en el tiempo $n\Delta t$ y la iteración q . Usando la notación anterior, el algoritmo de gradiente conjugado que usamos para resolver el problema de dimensión finita (3.31) es como sigue:

- Suponer que se tiene un valor de inicio

$$\mathbf{u}_0 = \{u_0^n\}_{n=1}^N \text{ es dado en } \mathcal{U}_{ad}^{\Delta t} = \mathbb{R}^N \text{ (} \mathbf{u}_0 = \mathbf{0} \text{ por ejemplo)}. \quad (3.46)$$

- Calcular $\{\mathbf{y}_0^n\}_{n=0}^N = \{\{\mathbf{y}_{i0}^n\}_{i=1}^3\}_{n=0}^N$ y $\{\mathbf{p}_0^n\}_{n=1}^{N+1} = \{\{\mathbf{p}_{i0}^n\}_{i=1}^3\}_{n=1}^{N+1}$, resolviendo los siguientes dos problemas

$$\left\{ \begin{array}{l} \mathbf{y}_0^0 = \mathbf{y}_0, \\ \text{para } n = 1, \dots, N \text{ resolver} \\ \Gamma \frac{\mathbf{y}_0^n - \mathbf{y}_0^{n-1}}{\Delta t} + K \mathbf{y}_0^n + \begin{pmatrix} \sin y_{10}^{n-1} \\ \sin y_{20}^{n-1} \\ \sin y_{30}^{n-1} \end{pmatrix} = \begin{pmatrix} i_1 + u_0^n \\ i_2 \\ i_3 \end{pmatrix}, \end{array} \right. \quad (3.47)$$

y entonces

$$\left\{ \begin{array}{l} \Gamma \mathbf{p}^{N+1} = k(\mathbf{y}^N - \mathbf{y}_T), \\ \Gamma \frac{\mathbf{p}^N - \mathbf{p}^{N+1}}{\Delta t} + K \mathbf{p}^N = \mathbf{0}, \\ \text{para } n = N-1, \dots, 1 \text{ resolver} \\ \Gamma \frac{\mathbf{p}_0^n - \mathbf{p}_0^{n+1}}{\Delta t} + K \mathbf{p}_0^n + \begin{pmatrix} \cos y_{10}^n & 0 & 0 \\ 0 & \cos y_{20}^n & 0 \\ 0 & 0 & \cos y_{30}^n \end{pmatrix} \mathbf{p}_0^{n+1} = \mathbf{0}. \end{array} \right. \quad (3.48)$$

- Hacer

$$\mathbf{g}_0 = \{g_0^n\}_{n=1}^N = \{u_0^n + p_{10}^n\}_{n=1}^N. \quad (3.49)$$

- Si

$$\frac{\Delta t \sum_{n=1}^N |g_0^n|^2}{\max[1, \Delta t \sum_{n=1}^N |u_0^n|^2]} \leq tol^2,$$

tomar $\mathbf{u}^{\Delta t} = \mathbf{u}_0$; en otro caso, fijar

$$\mathbf{w}_0 = \mathbf{g}_0. \quad (3.50)$$

Para $q \geq 0$, \mathbf{u}_q , \mathbf{g}_q y \mathbf{w}_q conocidos, los dos últimos diferentes de $\mathbf{0}$, calcular \mathbf{u}_{q+1} y, de ser necesario \mathbf{w}_{q+1} como sigue:

- Resolver

$$\begin{cases} \rho_q \in \mathbb{R}, \\ J^{\Delta t}(\mathbf{u}_q - \rho_q \mathbf{w}_q) \leq J^{\Delta t}(\mathbf{u}_q - \rho \mathbf{w}_q), \forall \rho \in \mathbb{R}. \end{cases} \quad (3.51)$$

- Calcular

$$\mathbf{u}_{q+1} = \mathbf{u}_q - \rho_q \mathbf{w}_q. \quad (3.52)$$

- Calcular $\{\mathbf{y}_{q+1}^n\}_{n=0}^N = \{\{\mathbf{y}_{i_{q+1}}^n\}_{i=1}^3\}_{n=0}^N$ y $\{\mathbf{p}_{q+1}^n\}_{n=1}^{N+1} = \{\{\mathbf{p}_{i_{q+1}}^n\}_{i=1}^3\}_{n=1}^{N+1}$, resolviendo los siguientes dos sistemas

$$\begin{cases} \mathbf{y}_{q+1}^0 = \mathbf{y}_0, \\ \text{para } n = 1, \dots, N \text{ resolver} \\ \Gamma \frac{\mathbf{y}_{q+1}^n - \mathbf{y}_{q+1}^{n-1}}{\Delta t} + K \mathbf{y}_{q+1}^n + \begin{pmatrix} \sin y_{1q+1}^{n-1} \\ \sin y_{2q+1}^{n-1} \\ \sin y_{3q+1}^{n-1} \end{pmatrix} = \begin{pmatrix} i_1 + u_{q+1}^n \\ i_2 \\ i_3 \end{pmatrix}, \end{cases} \quad (3.53)$$

y entonces

$$\begin{cases} \Gamma \mathbf{p}_{q+1}^{N+1} = k(\mathbf{y}_{q+1}^N - \mathbf{y}_T), \\ \Gamma \frac{\mathbf{p}_{q+1}^N - \mathbf{p}_{q+1}^{N+1}}{\Delta t} + K \mathbf{p}_{q+1}^N = \mathbf{0}, \\ \text{para } n = N-1, \dots, 1 \text{ resolver} \\ \Gamma \frac{\mathbf{p}_{q+1}^n - \mathbf{p}_{q+1}^{n+1}}{\Delta t} + K \mathbf{p}_{q+1}^n \\ + \begin{pmatrix} \cos y_{1q+1}^n & 0 & 0 \\ 0 & \cos y_{2q+1}^n & 0 \\ 0 & 0 & \cos y_{3q+1}^n \end{pmatrix} \mathbf{p}_{q+1}^{n+1} = \mathbf{0}. \end{cases} \quad (3.54)$$

- Hacer

$$\mathbf{g}_{q+1} = \{g_{q+1}^n\}_{n=1}^N = \{u_{q+1}^n + p_{1q+1}^n\}_{n=1}^N. \quad (3.55)$$

- Si

$$\frac{\Delta t \sum_{n=1}^N |g_{q+1}^n|^2}{\max[\Delta t \sum_{n=1}^N |g_0^n|^2, \Delta t \sum_{n=1}^N |u_{q+1}^n|^2]} \leq tol^2,$$

tomar $\mathbf{u}^{\Delta t} = \mathbf{u}_{q+1}$; en otro caso, calcular

$$\beta_q = \frac{\sum_{n=1}^N |g_{q+1}^n|^2}{\sum_{n=1}^N |g_q^n|^2} \quad (3.56)$$

y fijar

$$\mathbf{w}_{q+1} = \mathbf{g}_{q+1} + \beta_q \mathbf{w}_q. \quad (3.57)$$

- Hacer $q+1 \rightarrow q$ y regresar a (3.51).
- Fin del algoritmo.

No es difícil generalizar este algoritmo para que el control actúe sobre otra junta, o para que dos controles actúen sobre alguna combinación de juntas o para que tres controles actúen sobre las tres juntas. Aplicaremos estos algoritmos en las siguientes secciones para hacer transitar el sistema entre dos estados de equilibrio estables específicos, lo cual permitirá realizar operaciones de *Lectura/Escritura*, También se hará transitar el sistema desde algunos estados estables a algunos equilibrios inestables.

3.6 Operaciones de Memoria

El sistema de ecuaciones diferenciales (3.3) tiene varios conjuntos de desplazamiento de fase en estado de equilibrio, usando la cantidad de corriente y los parámetros de acoplamiento mencionados, como se muestra en la tabla 3.1. En muchos casos las fases de estado estable están cerca de los mínimos $2\pi n_k$ de los términos $-\cos\theta_k$, por lo que las fases en equilibrio se definen por sus desfases σ_k desde esos mínimos (Braiman & Neschke):

$$\theta_k = 2\pi n_k + \sigma_k. \tag{3.58}$$

n_1	n_2	n_3	$\sigma_1/2\pi$	$\sigma_2/2\pi$	$\sigma_3/2\pi$	θ_1	θ_2	θ_3
0	0	0(s)	0.1992	0.1187	-0.1552	1.2517	0.7458	-0.9752
1	0	0(u)	0.0908	0.3591	-0.4140	6.8539	2.2566	-2.6015
1	1	0(s)	0.1810	0.0338	-0.0515	7.4207	6.4958	-0.3236
2	0	0						
2	1	0(u)	0.1011	0.4539	-0.0125	13.2016	9.1355	-0.0786
2	1	0(s)	0.0661	0.1164	-0.0437	12.9821	7.0148	-0.2746
2	2	0(u)	0.1576	-0.1029	-0.5958	13.5568	11.9196	-3.7436
2	2	0(s)	0.1671	-0.0443	0.0333	13.6161	12.2884	0.2092

Tabla 3.1: Equilibrios para el sistema AMJJ. Los símbolos (u) y (s) significan que el equilibrio es inestable y estable, respectivamente.

La tabla 3.1 incluye todos los desfases de estados estables donde $n_1 \geq n_2 \geq n_3$ y $n_1 \leq 2$, y tres estados inestables: $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$, $\{2, 1, 0(u)\}$, $\{2, 2, 0(u)\}$. Para los estados estables, sin pérdida de generalidad, n_3 se definió como 0 ya que las localizaciones de los mínimos locales son simétricos al cambiar todas las fases por 2π . Para el conjunto de corrientes directas y los parámetros de acoplamiento mencionados, no existe estado estable cuando la fase de la primera junta es casi 2π (respectivamente 4π) mas grande que las fases en las otras dos juntas. Ver renglón donde $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$ (respectivamente $\{n_1, n_2, n_3\} = \{2, 0, 0(u)\}$). La ausencia de esos estados estables es crucial para el diseño de celdas de memoria, ya que, de acuerdo a Braiman & Neschke, para la aplicación en celdas de memoria es suficiente manipular el sistema para que transite entre dos estados de equilibrio, a saber, los estados estables $\{0, 0, 0(s)\}$ y $\{1, 1, 0(s)\}$.

El circuito de tres Juntas Rápidas Simples acopladas (RSJ por sus siglas en inglés) asociadas con el modelo (3.3) puede operar como una celda de memoria si un conjunto de operadores pueden provocar la transición del sistema entre estados estacionarios bien definidos y pueden emitir una señal que discrimine estados de la memoria. El valor n_k como se presenta en la ecuación (3.58), describirá la localización de la fase de la k -sima junta. Cuando las fases de las tres juntas estén en el mismo pozo de potencial sinusoidal, se considerará que el sistema está en el estado ‘0’, o sea el estado $\{n_1, n_2, n_3\} = \{0, 0, 0(s)\}$. Cuando las fases de la primera y la segunda junta saltan al siguiente pozo de potencial (cerca de 2π hacia arriba), la celda estará en el estado ‘1’, esto es $\{n_1, n_2, n_3\} = \{1, 1, 0(s)\}$. Estos dos estados corresponden a la primera y tercera filas de la tabla 3.1. En la siguiente subsección mostraremos cómo las operaciones básicas de memoria pueden definirse al hacer transitar el sistema entre esos estados como se muestra en la figura 3.1.

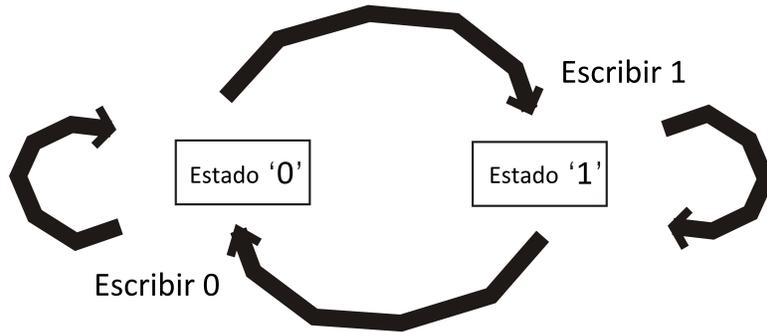


Figura 3.1: Los dos operadores ‘Escribir 0’ y ‘Escribir 1’ tienen que enviar el sistema al estado previsto desde cualquier estado inicial ‘0’ o ‘1’.

La figura 3.2 muestra las fases de las juntas cuando el sistema inicia con condiciones iguales a cero. Las fases alcanzan el estado estable ‘0’, es decir, el estado en donde todas las fases están cerca del mismo múltiplo de 2π . Por conveniencia, varias de las siguientes figuras están normalizadas por 2π . En esta sección se usan los mismos valores para los parámetros que en (3.4), (3.5) y $\Delta t = 1/500$, $\varepsilon = 1E - 6$, a menos que se diga lo contrario. Las figuras 3.3, 3.4 y 3.5 muestran el comportamiento del sistema cuando las condiciones iniciales son aproximaciones de los equilibrios inestables $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}, \{2, 1, 0(u)\}, \{2, 2, 0(u)\}$, respectivamente.

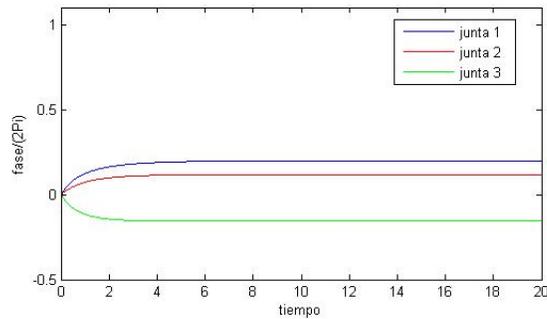


Figura 3.2: Series de tiempo de las fases escaladas por 2π , con condiciones iniciales cero.

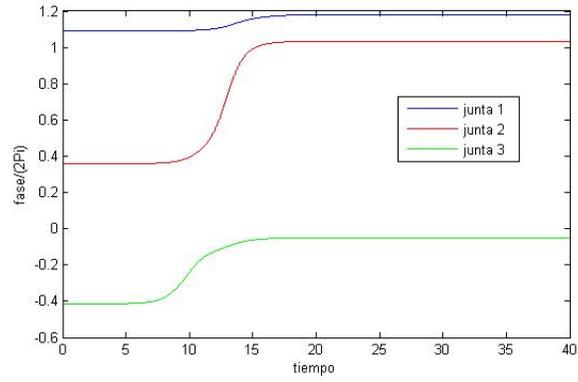


Figura 3.3: Evolución al estado $\{1, 1, 0(s)\}$ desde una aproximación del equilibrio $\{1, 0, 0(u)\}$.

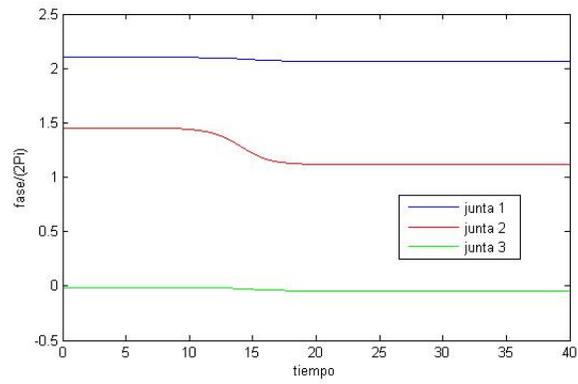


Figura 3.4: Evolución al estado $\{2, 1, 0(s)\}$ desde una aproximación del equilibrio $\{2, 1, 0(u)\}$.

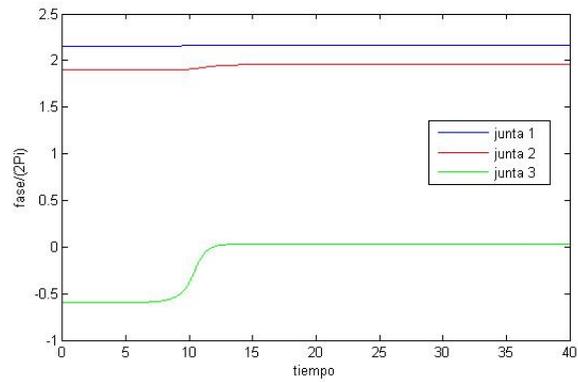


Figura 3.5: Evolución al estado $\{2, 2, 0(s)\}$ desde una aproximación del equilibrio $\{2, 2, 0(u)\}$.

3.6.1 Operación ‘Escribir 1’

Si un control que se aplica al sistema siempre produce una transición al estado ‘1’, independientemente de si estaba en el estado ‘1’ o en el estado ‘0’, entonces la operación puede ser considerada como escritura en memoria del ‘1’, o ‘Escribir 1’; el control es el operador. La configuración de equilibrio de las fases en el estado ‘1’ es tal que la primera y segunda fase saltaron al siguiente pozo de potencial (aproximadamente 2π mayor que la fase de la tercera junta). Para que la transición desde el estado ‘0’ hasta el estado ‘1’ se lleve a cabo exitosamente, la energía inyectada en el sistema por el control debe afectar sólo las fases de la primera y segunda juntas. Buscaremos un control que actúe en la primera junta para llevar a cabo la operación ‘Escribir 1’. Debido al acoplamiento entre la primera y segunda juntas, este control causará también un salto en la fase de la segunda junta. La fase de la tercera junta no debe cambiar significativamente. Las figuras 3.6, 3.7, 3.8 y 3.9 muestran, para varios valores de T , los controles y las fases de las juntas que hemos encontrado al aplicar el algoritmo (3.46)-(3.57) con el fin de ir del estado ‘0’ al estado ‘1’.

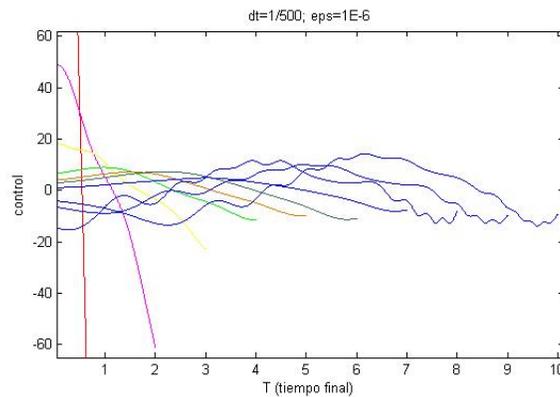


Figura 3.6: Controles para varios valores de T . La transición es desde el estado ‘0’ hasta el estado ‘1’.

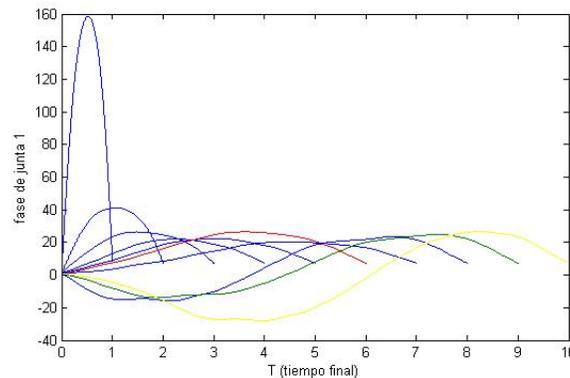


Figura 3.7: Fases de la junta 1 para diferentes valores de T , asociadas con los controles de la figura 3.6.

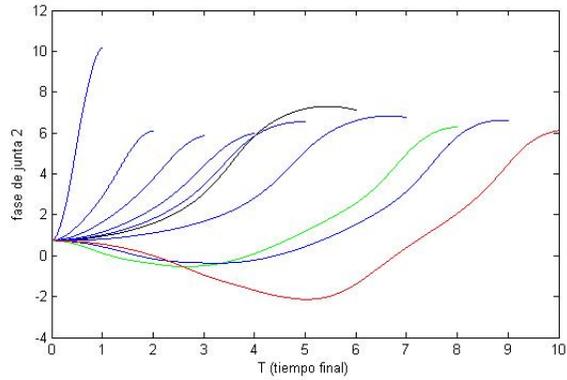


Figura 3.8: Fases de la junta 2 para diferentes valores de T , asociadas con los controles de la figura 3.6.

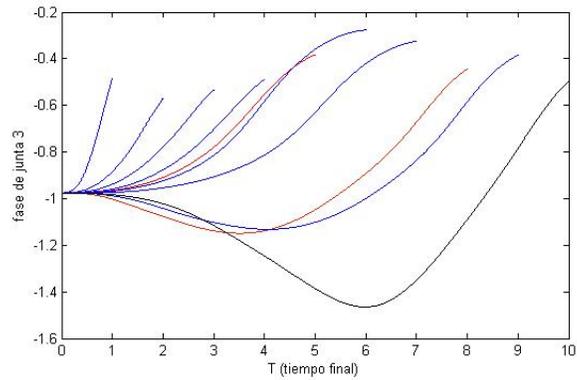


Figura 3.9: Fases de la junta 3 para diferentes valores de T , asociadas con los controles de la figura 3.6.

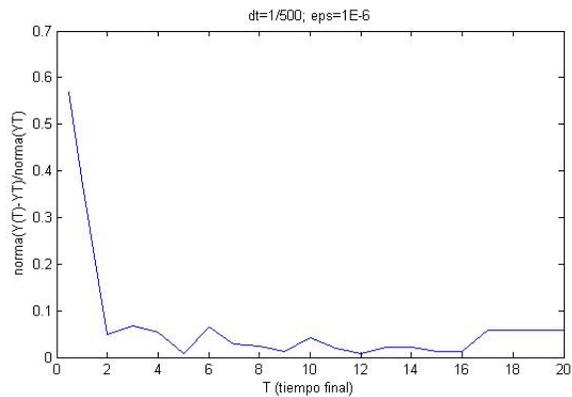


Figura 3.10: Norma relativa del estado final aproximado cuando se transita del estado ‘0’ al estado ‘1’ usando los controles de la figura 3.6.

La figura 3.10 muestra el comportamiento de $\|\mathbf{y}^N - \mathbf{y}_T\|/\|\mathbf{y}_T\|$ cuando T se incrementa de 1 a 20. Para el caso específico del control correspondiente a $T = 7$, y extendido como $u = 0$ hasta $T = 14$, el cual se muestra en la figura 3.11, podemos ver en la figura 3.12 el comportamiento del sistema al iniciar en el estado ‘0’ y el control actuando en la primera junta. El comportamiento de las fases en la figura 3.12 es el esperado ya que el estado ‘1’ es estable, pero cuando el control es aplicado al sistema estando éste en el estado ‘1’, se va al estado $\{2,1,0(s)\}$ como en la figura 3.13, en lugar de permanecer en el estado ‘1’, una condición necesaria para que el control sea útil como un operador ‘Escribir 1’. Lo mismo sucede para cualquier control en la figura 3.6.

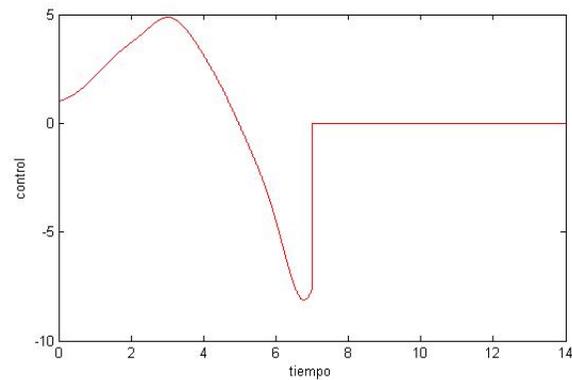


Figura 3.11: Control para $T = 7$ de la figura 3.6, extendido como $u = 0$.

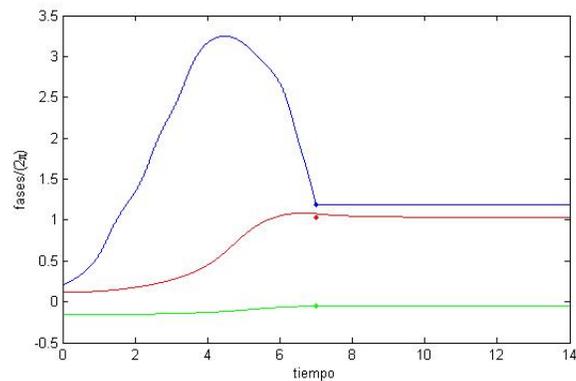


Figura 3.12: Comportamiento de las fases cuando se inicia en el estado ‘0’ y se aplica en la primera junta el control de la figura 3.11. El sistema transita al estado ‘1’.

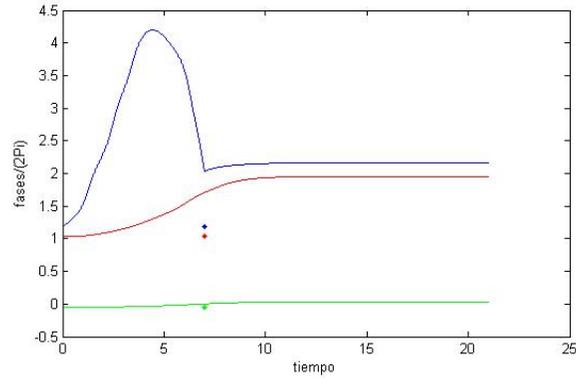


Figura 3.13: Comportamiento de las fases cuando se inicia en el estado ‘1’ y se aplica en la primera junta el control de la figura 3.11. El sistema transita al estado $\{2, 2, 0(s)\}$.

Gracias al hecho de que el estado ‘0’ y el estado ‘1’ son estables y a que existe un equilibrio inestable entre ellos (equilibrio $\{1, 0, 0(u)\}$ en la tabla 3.1), podemos tratar de alcanzar algún estado en la cuenca de atracción del estado ‘1’, pero suficientemente alejado de éste para garantizar que cuando el mismo control sea aplicado al sistema pero iniciando en el estado ‘1’, éste permanezca en el mismo estado ‘1’. Por ejemplo, para $T = 20$, el control óptimo que tiene este efecto se muestra en la figura 3.14 ($Y_T = 2\pi\{0.2489, 0.5053, -0.1136\}$). La transición del estado ‘0’ al estado ‘1’ y la permanencia en el estado ‘1’ se ilustra en la figura 3.15.

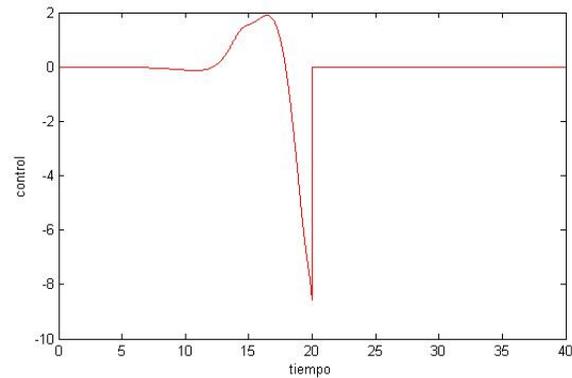


Figura 3.14: Control útil como operador ‘Escribir 1’ cuando $T = 20$.

Así que el control de la figura 3.14 puede verse como un operador ‘Escribir 1’ (para el caso $T = 20$) ya que lleva al sistema al estado ‘1’, independientemente del estado de inicio.

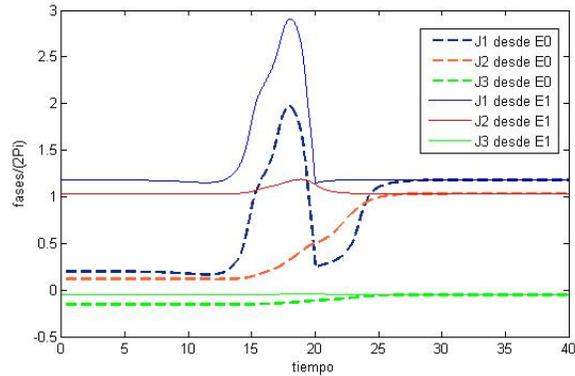


Figura 3.15: Transición del sistema al estado ‘1’ desde los dos estados ‘0’ y ‘1’ usando el control de la figura 3.14.

3.6.2 Operación ‘Escribir 0’

La operación de memoria ‘Escribir 0’ o reset se implementa controlando la energía en la tercera junta. En este caso se debe dar la transición desde el estado ‘1’ de regreso al estado ‘0’, y además, utilizando el mismo control e iniciando en el estado ‘0’ el sistema debe permanecer, después de un transitorio, en el estado ‘0’. En este caso, el control usado se llama un operador ‘Escribir 0’.

Si el sistema está en el estado ‘1’, una vez que la fase de la tercera junta se mueva al siguiente pozo de potencial, todas las fases estarán en el mismo nivel. Las fases del nuevo estado estable son exactamente 2π mas grandes que las alcanzadas después de iniciar con condiciones cero y sin energía adicional en las juntas. El nuevo estado es $\{n_1, n_2, n_3\} = \{1, 1, 1(s)\}$, el cual, por definición es equivalente al estado $\{0, 0, 0(s)\}$. Las figuras 3.16-3.19 muestran los controles y las fases de las juntas para varios valores de T . La figura 3.20 muestra el comportamiento de $\|\mathbf{y}^N - \mathbf{y}_T\|/\|\mathbf{y}_T\|$ cuando T incrementa de 1 a 20. Un control específico que funciona como operador de escritura ‘0’ se muestra en la figura 3.21. La figura 3.22 muestra la transición del estado ‘1’ al estado ‘0’ y cómo el sistema permanece en el estado ‘1’.

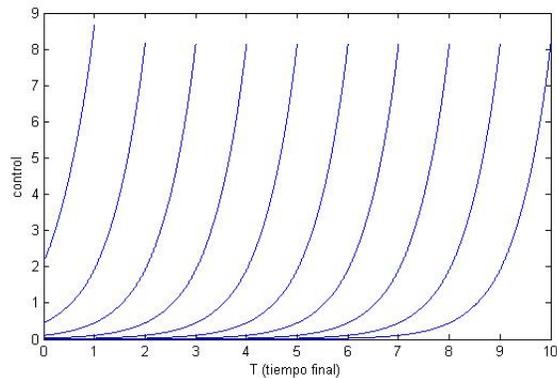


Figura 3.16: Controles para varios valores de T . Controlando la tercera junta el sistema va del estado ‘1’ al estado ‘0’.

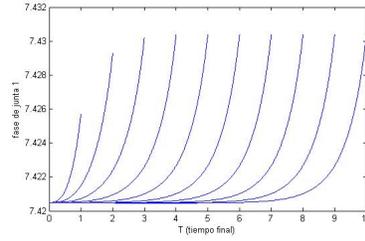


Figura 3.17: Fases de la junta 1 para diferentes valores de T , asociadas con los controles de la figura 3.16.

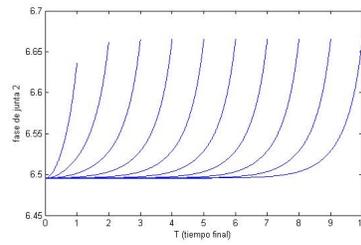


Figura 3.18: Fases de la junta 2 para diferentes valores de T , asociadas con los controles de la figura 3.16.

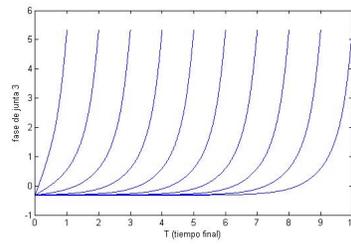


Figura 3.19: Fases de la junta 3 para diferentes valores de T , asociadas con los controles de la figura 3.16.

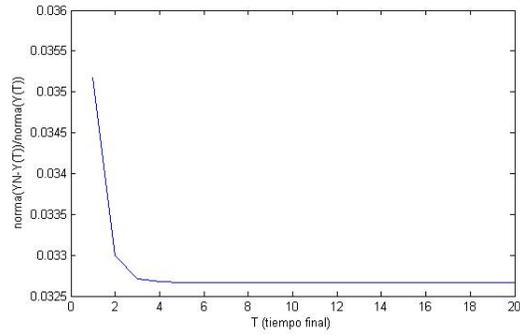


Figura 3.20: Norma relativa del estado final.

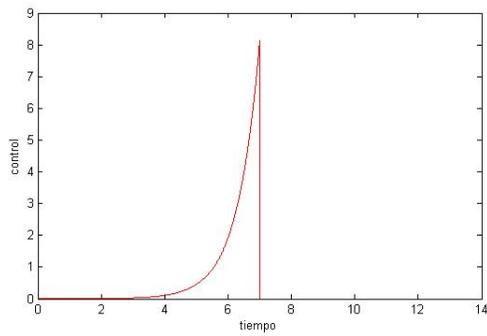


Figura 3.21: Control actuando como operador 'Escribir 0'.

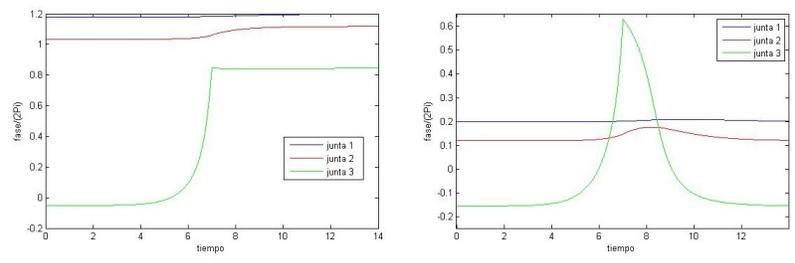


Figura 3.22: Transición del estado '1' al estado '0' (izquierda); Inicio y permanencia en el estado '0' (derecha).

3.6.3 Operación Leer

Siguiendo a Y. Braiman y B. Neschke [33], para leer el estado de la memoria, usamos un control que pueda aplicarse a una de las juntas. Una condición para tener una operación de *lectura* exitosa es que la aplicación del control de *lectura* proporcione diferentes resultados en el circuito, dependiendo de si el circuito inicialmente estaba en el estado ‘0’ o en el estado ‘1’. El mismo control (figura 3.14) que usamos para la operación ‘Escribir 1’ (figura 3.15), aplicado a la primera junta, puede servir como nuestro comando de *lectura*. Los datos de *lectura* (0 ó 1) se tomarán de la segunda junta. Cuando el estado inicial del circuito es ‘1’, casi no habrá respuesta de fase de la segunda junta al control aplicado. Por el contrario, cuando el estado inicial del circuito de memoria es ‘0’, el control produce un salto de 2π en la fase de la segunda junta. Consecuentemente, la segunda junta muestra un correspondiente pico de voltaje. Así que la respuesta de la segunda junta depende del estado inicial del circuito. Sin embargo, como es común para operaciones de lectura destructivas, después de que se aplique el comando de *lectura*, el estado del circuito estará siempre en el estado ‘1’.

3.7 Alcanzando un estado de equilibrio inestable

En las secciones previas estuvimos interesados en alcanzar estados de equilibrio estables del sistema (3.3) iniciando desde otros estados de equilibrio estables. Para ese propósito fué suficiente controlar sólo sobre una junta. En esta sección estamos interesados en alcanzar estados inestables a partir de estados estables. En este caso no es suficiente controlar sólo sobre una junta. Ya que podemos controlar sobre 1, 2 o 3 juntas, resumiremos los resultados siguiendo este orden. Todos los resultados mostrados en esta sección fueron obtenidos usando los valores $\Delta t = 1/500$, $\varepsilon = 1E - 6$ y los valores en (3.4)-(3.5). En todas las gráficas los colores azul, rojo y verde están asociados con las juntas 1, 2 y 3 respectivamente, y se incluyen tres puntos, azul, rojo y verde, para ilustrar las componentes del equilibrio que estamos tratando de alcanzar.

3.7.1 Controlando con una junta

Aquí el único caso exitoso fue aquel en el que se alcanzó el equilibrio inestable $\{2, 1, 0(u)\}$ desde el estado estable cercano $\{2, 1, 0(s)\}$ cuyos valores específicos son $\{12.9821, 7.0148, -0.2746\}$ y $\{13.2016, 9.1355, -0.0786\}$ respectivamente. Los resultados para las fases se muestran en la figura 3.23 y el control que se usó, actuando en la primera junta, se muestra en la figura 3.24. El control se obtuvo con $T = 10$ y fijamos $u = 0$ una vez que el equilibrio inestable se alcanzó .

En general, cuando se controla sólo una junta, únicamente se alcanza la respectiva componente del vector objetivo, como se muestra en las siguientes figuras. En la figura 3.25 se muestra el intento de transitar el sistema desde el estado ‘0’ hasta el estado inestable $\{1, 0, 0(u)\}$, aplicando el control en la primera junta. No se alcanza el estado que se busca y al dejar de controlar, el sistema transita al estado estable $\{1, 1, 0(s)\}$ (estado ‘1’). En la figura 3.26 tratamos otra vez de transitar el sistema desde el estado ‘0’ hasta el estado inestable $\{1, 0, 0(u)\}$, pero ahora aplicando el control en la tercera junta. Tampoco se pudo alcanzar el valor objetivo para la primera componente y al dejar de controlar, el sistema evolucionó al estado estable $\{2, 1, 0(s)\}$.

La figura 3.27 muestra los resultados de tratar de alcanzar el estado inestable $\{1, 0, 0(u)\}$ pero ahora desde el estado ‘1’, controlando en la primera junta. Después de un transitorio el sistema

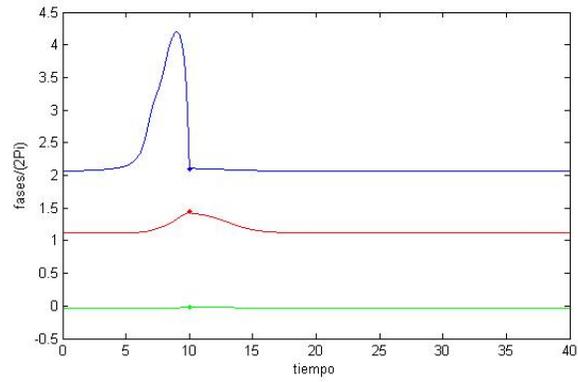


Figura 3.23: Alcanzando el equilibrio inestable $\{2, 1, 0(u)\}$ desde el estable $\{2, 1, 0(s)\}$.

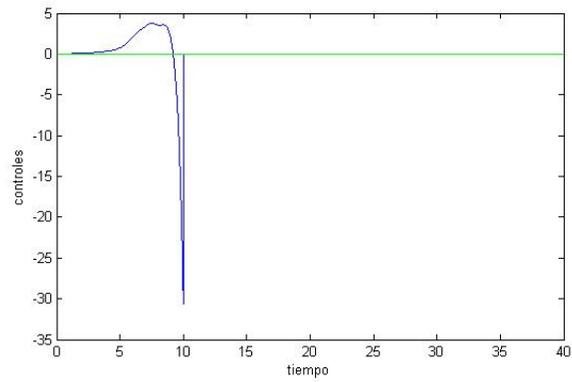


Figura 3.24: Control aplicado en la primera junta para alcanzar el equilibrio inestable $\{2, 1, 0(u)\}$ desde el estable cercano $\{2, 1, 0(s)\}$.

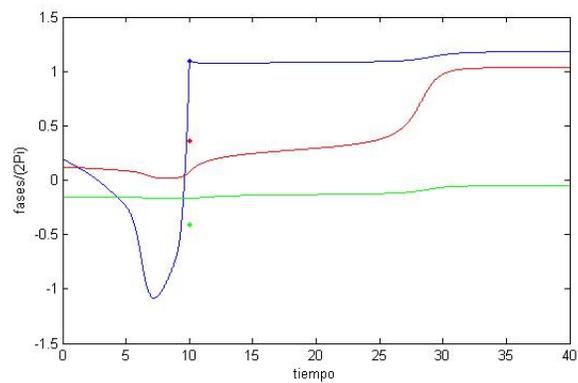


Figura 3.25: Intento de transitar el sistema del estado '0' al estado inestable $\{1, 0, 0(u)\}$, controlando en la junta 1.

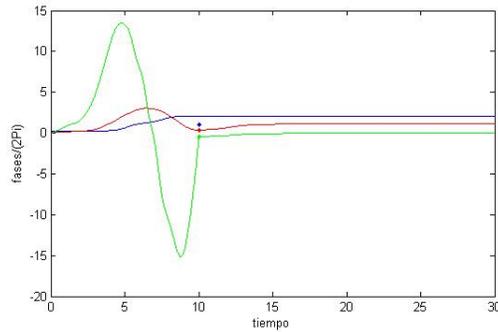


Figura 3.26: Intento de transitar el sistema del estado ‘0’ al estado inestable $\{1, 0, 0(u)\}$, controlando en la junta 3.

permanece en el estado ‘1’.

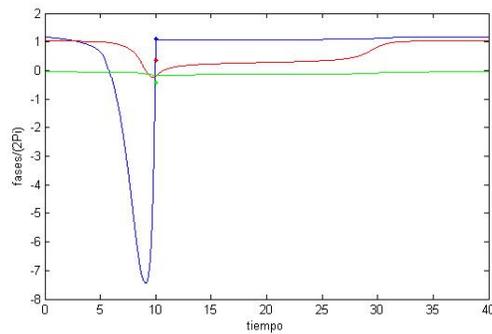


Figura 3.27: Intento de trasitar el sistema desde el estado ‘1’ hasta el estado inestable $\{1, 0, 0(u)\}$. Obsérvese que permanece en el estado ‘1’.

3.7.2 Controlando con dos juntas

En general, cuando controlamos sobre dos juntas, se alcanzan los respectivos valores del vector objetivo, pero el que se alcance o no el valor objetivo asociado con la junta sobre la que no hay control, depende de cuál par de juntas y puntos de equilibrio se seleccionen. Para comparar con los resultados de la sección previa presentamos los resultados para casos similares. En la figura 3.28 presentamos la transición del sistema desde el estado estable $\{0, 0, 0(s)\}$ al estado inestable $\{1, 0, 0(s)\}$ cuando se controla sobre las juntas 1 y 3. En las figuras 3.29 y 3.30 mostramos la transición del sistema desde el estado estable ‘1’, $\{1, 1, 0(s)\}$, al estado inestable $\{1, 0, 0(u)\}$ cuando se controlan las juntas 1 y 2, y 2 y 3 respectivamente.

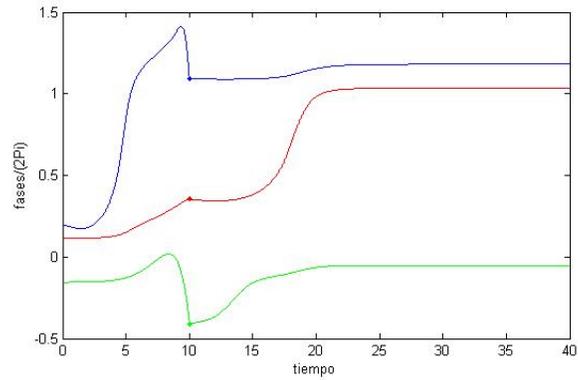


Figura 3.28: Transición del sistema del estado estable $\{0, 0, 0(s)\}$ al estado inestable $\{1, 0, 0(u)\}$ cuando se controlan las juntas 1 y 3.

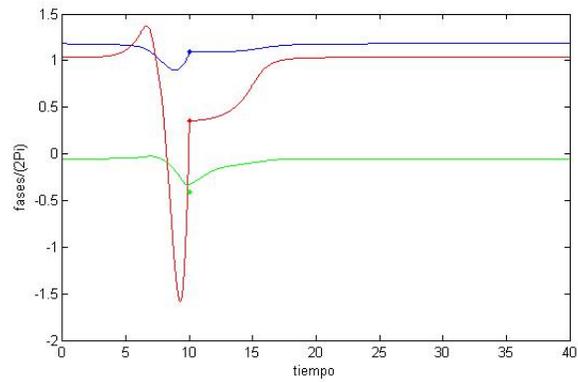


Figura 3.29: Transición del sistema desde el estado estable $\{1, 1, 0(s)\}$ tratando de alcanzar el estado inestable $\{1, 0, 0(u)\}$, controlando las juntas 1 y 2.

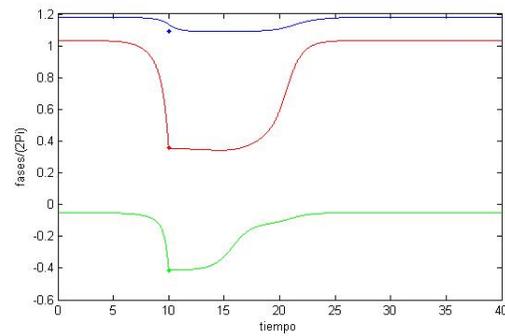


Figura 3.30: Transición del sistema desde el estado estable $\{1, 1, 0(s)\}$ tratando de alcanzar el estado inestable $\{1, 0, 0(u)\}$, controlando las juntas 2 y 3.

3.7.3 Controlando con tres juntas

Cuando controlamos con las tres juntas alcanzamos cualquier equilibrio inestable con un error relativo pequeño y en pocas iteraciones. En la figura 3.31 mostramos la transición del sistema desde el estado estable $\{1, 1, 0(s)\}$ al estado inestable $\{1, 0, 0(u)\}$ cuando se controlan las tres juntas.

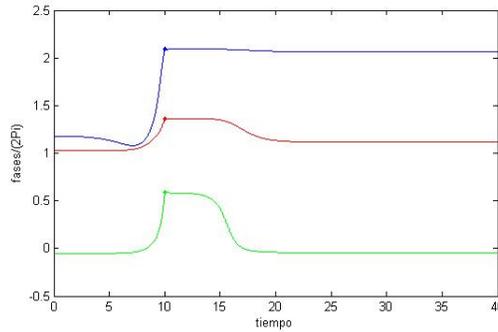


Figura 3.31: Transición del sistema desde el estado estable $\{1, 1, 0(s)\}$ al estado inestable $\{1, 0, 0(s)\}$ cuando se controlan las tres juntas.

3.8 Conclusiones

Se ha propuesto una formulación de control óptimo que permite conducir la evolución de las fases del modelo AMJJ desde un estado de equilibrio a otro. Las condiciones de optimalidad se encontraron por medio de cálculo variacional usando un análisis de perturbación. Como el modelo AMJJ es no lineal se utilizó el algoritmo de gradiente conjugado general no cuadrático para encontrar una aproximación al control óptimo. En este tipo de problemas es necesario utilizar técnicas efectivas de búsqueda de línea (line search) en cada iteración debido a que no es posible encontrar el tamaño de paso de forma exacta. Nuestra elección fue el método de Newton. Este enfoque numérico produce excelentes resultados como se muestra en los ejemplos, que incluyen:

- La transición entre dos estados de equilibrio estables específicos, que a su vez permiten encontrar controles que sirvan como operadores de memoria de *lectura* y *escritura*. Los resultados numéricos muestran que para lograr esta transición fue suficiente controlar sobre una sola junta.
- La transición desde estados estables a estados inestables. En este caso es necesario controlar al menos sobre dos juntas y los resultados dependen de cuales estados de equilibrio y cuáles juntas se consideren.

Capítulo 4

Estabilización del AMJJ alrededor de un equilibrio inestable

4.1 El enfoque de controlabilidad

Como se explicó en el capítulo 3, las figuras 3.3, 3.4 y 3.5 muestran el comportamiento del sistema (3.3) cuando las condiciones iniciales son aproximaciones de los equilibrios inestables $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}, \{2, 1, 0(u)\}, \{2, 2, 0(u)\}$, respectivamente. En este capítulo nuestro objetivo es estabilizar el sistema modelado por (3.3), alrededor de un equilibrio inestable, controlando por medio de las juntas, i.e., deseamos mantener los valores de las fases de las figuras 3.3, 3.4 y 3.5 casi constantes a través del tiempo. El valor constante debe ser el valor inicial en cada caso. El enfoque que se considera aquí es el clásico, a saber,

- (a) Linealizar el modelo (3.3) en la vecindad de un equilibrio (inestable) del sistema.
- (b) Calcular un control óptimo para el modelo linealizado.
- (c) Aplicar el control anterior al sistema no lineal.

Consideremos un estado inestable θ de (3.3) y una variación “pequeña” $\delta\theta$ de θ ; la perturbación $\delta\phi$ del estado de equilibrio θ satisface aproximadamente el modelo lineal siguiente:

$$\begin{cases} \gamma_1 \frac{d\delta\phi_1}{dt} + \kappa_1(\delta\phi_1 - \delta\phi_2) + \delta\phi_1 \cos\theta_1 = 0, \\ \gamma_2 \frac{d\delta\phi_2}{dt} + \kappa_1(\delta\phi_2 - \delta\phi_1) + \kappa_2(\delta\phi_2 - \delta\phi_3) + \delta\phi_2 \cos\theta_2 = 0, \\ \gamma_3 \frac{d\delta\phi_3}{dt} + \kappa_2(\delta\phi_3 - \delta\phi_2) + \delta\phi_3 \cos\theta_3 = 0, \\ \delta\phi(0) = \delta\theta. \end{cases} \quad (4.1)$$

Al menos uno de los valores propios del jacobiano del sistema (4.1) es positivo. Esto significa que el sistema puede presentar un fenómeno de explosión (en tiempo infinito). Las figuras 4.1, 4.2 y 4.3 muestran la solución de (4.1) para los tres equilibrios inestables $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}, \{2, 1, 0(u)\}, \{2, 2, 0(u)\}$ incluidos en la tabla 3.1, respectivamente. Los valores usados para la perturbación inicial $\delta\theta$ fueron $\delta\theta = [1E - 2, 1E - 2, 1E - 2]$.

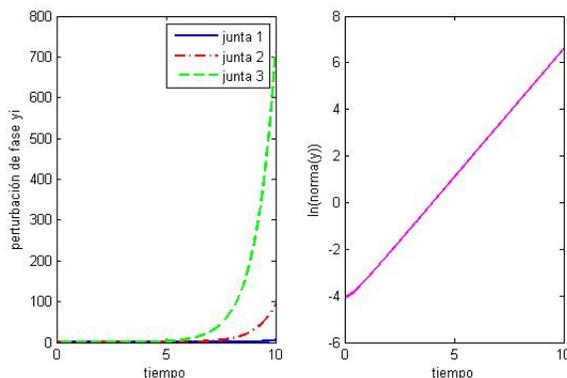


Figura 4.1: Solución $\mathbf{y} = \delta\phi$ de (4.1) para el equilibrio inestable $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$ y $\delta\theta = [1E - 2, 1E - 2, 1E - 2]$.

Es claro que el modelo (4.1) ya no es válido si $\delta\theta$ es muy grande pero la idea detrás de considerar el modelo linealizado (4.1) es usarlo para calcular un control que prevenga que $\delta\phi$ se haga demasiado grande (y de ser posible llevarlo a cero), esperando que el control calculado

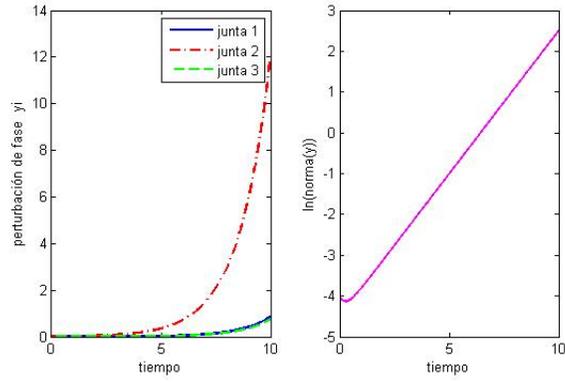


Figura 4.2: Solución $y = \delta\phi$ de (4.1) para el equilibrio inestable $\{n_1, n_2, n_3\} = \{2, 1, 0(u)\}$ y $\delta\theta = [1E - 2, 1E - 2, 1E - 2]$.

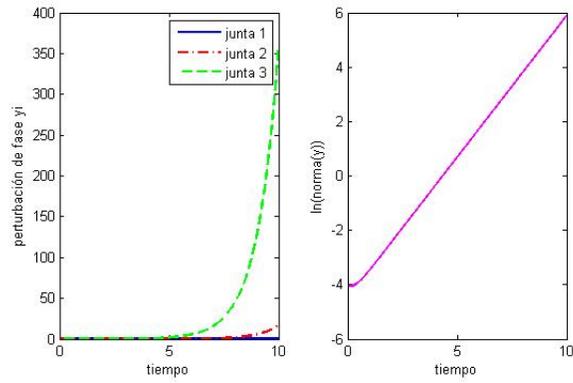


Figura 4.3: Solución $y = \delta\phi$ de (4.1) para el equilibrio inestable $\{n_1, n_2, n_3\} = \{2, 2, 0(u)\}$ y $\delta\theta = [1E - 2, 1E - 2, 1E - 2]$.

también estabilice el sistema no lineal original:

$$\begin{cases} \gamma_1 \frac{d\phi_1}{dt} + \kappa_1(\phi_1 - \phi_2) + \sin \phi_1 = I_1, \\ \gamma_2 \frac{d\phi_2}{dt} + \kappa_1(\phi_2 - \phi_1) + \kappa_2(\phi_2 - \phi_3) + \sin \phi_2 = I_2, \\ \gamma_3 \frac{d\phi_3}{dt} + \kappa_2(\phi_3 - \phi_2) + \sin \phi_3 = I_3, \\ \phi(0) = \theta + \delta\theta. \end{cases} \quad (4.2)$$

Los términos fuente en el lado derecho de (4.2) son de la forma $I_l = i_l + v_l$, con i_l corrientes directas cuyos valores ya fueron definidos en el capítulo 3 y v_l podría ser una energía adicional. En la figura 4.4 se muestra el comportamiento del sistema no lineal (4.2) cuando no se usa control ($v_l = 0, l = 1, 2, 3$) y se considera una perturbación inicial $\delta\theta = [1E - 2, 1E - 2, 1E - 2]$ del equilibrio inestable $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. El sistema transita al estado estable $\{1, 1, 0(s)\}$.

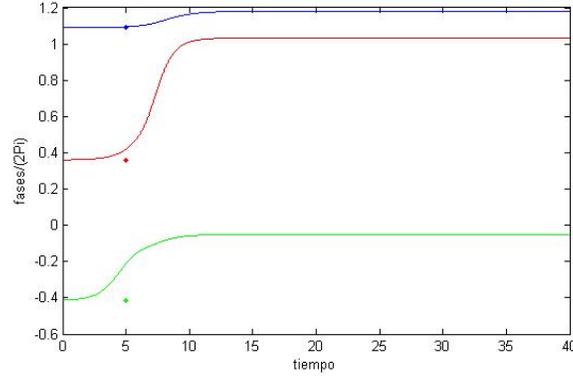


Figura 4.4: Solución ϕ de (4.2) cuando no se controla y la perturbación inicial es $\delta\theta = [1E - 2, 1E - 2, 1E - 2]$ del equilibrio inestable $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$.

4.2 El problema de control lineal

La variante con control del sistema (4.1) que consideramos (controlando por medio de las tres juntas) se define por

$$\begin{cases} \gamma_1 \frac{d\delta\phi_1}{dt} + \kappa_1(\delta\phi_1 - \delta\phi_2) + \delta\phi_1 \cos \theta_1 = v_1, \\ \gamma_2 \frac{d\delta\phi_2}{dt} + \kappa_1(\delta\phi_2 - \delta\phi_1) + \kappa_2(\delta\phi_2 - \delta\phi_3) + \delta\phi_2 \cos \theta_2 = v_2, \\ \gamma_3 \frac{d\delta\phi_3}{dt} + \kappa_2(\delta\phi_3 - \delta\phi_2) + \delta\phi_3 \cos \theta_3 = v_3, \\ \delta\phi(0) = \delta\theta. \end{cases} \quad (4.3)$$

Usando la notación $\mathbf{y} = \delta\phi$, estabilizaremos (de ser posible)

$$\begin{cases} \gamma_1 \frac{dy_1}{dt} + \kappa_1(y_1 - y_2) + y_1 \cos \theta_1 = v_1, \\ \gamma_2 \frac{dy_2}{dt} + \kappa_1(y_2 - y_1) + \kappa_2(y_2 - y_3) + y_2 \cos \theta_2 = v_2, \\ \gamma_3 \frac{dy_3}{dt} + \kappa_2(y_3 - y_2) + y_3 \cos \theta_3 = v_3, \\ \mathbf{y}(0) = \delta\theta. \end{cases} \quad (4.4)$$

utilizando la formulación de control siguiente:

$$\begin{cases} \mathbf{u} \in \mathcal{U}, \\ J(\mathbf{u}) \leq J(\mathbf{v}), \forall \mathbf{v} \in \mathcal{U}, \end{cases} \quad (4.5)$$

donde

$$J(\mathbf{v}) = \frac{1}{2} \int_0^T |\mathbf{v}|^2 dt + \frac{k_1}{2} \int_0^T |\mathbf{y}|^2 dt + \frac{k_2}{2} \|\mathbf{y}(T)\|^2, \quad (4.6)$$

con $|\mathbf{y}|^2 = |y_1|^2 + |y_2|^2 + |y_3|^2$.

4.3 Condiciones de optimalidad y solución de gradiente conjugado para el problema (4.5)

4.3.1 Generalidades y Sinopsis

Denotemos por $DJ(\mathbf{v})$ el diferencial de J en $\mathbf{v} \in \mathcal{U} = L^2(0, T; 3) \equiv (L^2(0, T))^3$. Ya que \mathcal{U} es un espacio de Hilbert para el producto escalar definido por

$$(\mathbf{v}, \mathbf{w})_{\mathcal{U}} = \int_0^T \mathbf{v}(t) \cdot \mathbf{w}(t) dt,$$

la acción $\langle DJ(\mathbf{v}), \mathbf{w} \rangle$ de $DJ(\mathbf{v})$ en $\mathbf{w} \in \mathcal{U}$ se puede escribir como

$$\langle DJ(\mathbf{v}), \mathbf{w} \rangle = \int_0^T DJ(\mathbf{v})(t) \cdot \mathbf{w}(t) dt, \quad \forall \mathbf{v}, \mathbf{w} \in \mathcal{U},$$

con $DJ(\mathbf{v})(t) \in \mathcal{U}$. Si \mathbf{u} es la solución del problema (4.5), se caracteriza [usando argumentos de convexidad (ver, e.g., [37])] por

$$DJ(\mathbf{u}) = \mathbf{0}. \quad (4.7)$$

Ya que la función costo J es cuadrática y el modelo de estado (4.1) es lineal, $DJ(\mathbf{v})$ es de hecho una función afín de \mathbf{v} , implicando junto con (4.7) que \mathbf{u} es la solución de una ecuación lineal en el espacio de control \mathcal{U} . En forma abstracta, la ecuación (4.7) se puede escribir como

$$DJ(\mathbf{u}) - DJ(\mathbf{0}) = -DJ(\mathbf{0}),$$

y por las propiedades (que se necesitan probar) del operador $\mathbf{v} \rightarrow DJ(\mathbf{v}) - DJ(\mathbf{0})$, el problema $DJ(\mathbf{u}) = \mathbf{0}$ se podría resolver por un algoritmo de gradiente conjugado (caso cuadrático, ver Apéndice A) operando en el espacio \mathcal{U} . La implementación práctica de este algoritmo requiere del conocimiento explícito de $DJ(\mathbf{v})$.

4.3.2 Calculando $DJ(\mathbf{v})$

Calculamos el diferencial $DJ(\mathbf{v})$ de la función costo J en \mathbf{v} suponiendo que $\mathcal{U} = L^2(0, T; \mathbb{R}^3)$. Para lograr esa meta emplearemos un análisis de perturbación. Consideremos así una perturbación $\delta\mathbf{v}$ de la variable de control \mathbf{v} . Tenemos entonces,

$$\left\{ \begin{array}{l} \delta J(\mathbf{v}) = \int_0^T DJ(\mathbf{v}) \cdot \delta\mathbf{v} dt \\ = \int_0^T \mathbf{v} \cdot \delta\mathbf{v} dt + k_1 \int_0^T \mathbf{y} \cdot \delta\mathbf{y} dt + k_2 \mathbf{y}(T) \cdot \delta\mathbf{y}(T), \end{array} \right. \quad (4.8)$$

donde en (4.8):

(i) Denotamos por $\mathbf{a} \cdot \mathbf{b}$ el producto interno de dos vectores \mathbf{a} y \mathbf{b} .

(ii) La función $\delta\mathbf{y} = \{\delta y_1, \delta y_2, \delta y_3\}$ es la solución del problema de valores iniciales siguiente, obtenido por perturbación de (4.4):

$$\left\{ \begin{array}{l} \gamma_1 \frac{d\delta y_1}{dt} + \kappa_1(\delta y_1 - \delta y_2) + C_1 \delta y_1 = \delta v_1 \text{ en } (0, T), \\ \gamma_2 \frac{d\delta y_2}{dt} + \kappa_1(\delta y_2 - \delta y_1) + \kappa_2(\delta y_2 - \delta y_3) + C_2 \delta y_2 = \delta v_2 \text{ en } (0, T), \\ \gamma_3 \frac{d\delta y_3}{dt} + \kappa_2(\delta y_3 - \delta y_2) + C_3 \delta y_3 = \delta v_3 \text{ en } (0, T), \\ \delta\mathbf{y}(0) = \mathbf{0}, \end{array} \right. \quad (4.9)$$

donde $C_1 = \cos \theta_1$, $C_2 = \cos \theta_2$, $C_3 = \cos \theta_3$. En forma matricial (4.9) queda como:

$$\left\{ \begin{array}{l} \Gamma \frac{d}{dt} \delta\mathbf{y} + K \delta\mathbf{y} + C \delta\mathbf{y} = \begin{pmatrix} \delta v_1 \\ \delta v_2 \\ \delta v_3 \end{pmatrix} \text{ en } (0, T), \\ \delta\mathbf{y}(0) = \mathbf{0}. \end{array} \right. \quad (4.10)$$

$$\text{con } C = \begin{pmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{pmatrix}, \quad \Gamma = \begin{pmatrix} \gamma_1 & 0 & 0 \\ 0 & \gamma_2 & 0 \\ 0 & 0 & \gamma_3 \end{pmatrix} \text{ y}$$

$$K = \begin{pmatrix} \kappa_1 & -\kappa_1 & 0 \\ -\kappa_1 & \kappa_1 + \kappa_2 & -\kappa_2 \\ 0 & -\kappa_2 & \kappa_2 \end{pmatrix}.$$

Introducimos ahora la función vectorial $\mathbf{p} = \{p_1, p_2, p_3\}$ que suponemos diferenciable sobre $(0, T)$; multiplicando por \mathbf{p} ambos lados de la ecuación diferencial en (4.10), e integrando sobre $(0, T)$ obtenemos:

$$\int_0^T \Gamma \frac{d}{dt} \delta\mathbf{y} \cdot \mathbf{p} dt + \int_0^T (K + C) \delta\mathbf{y} \cdot \mathbf{p} dt = \int_0^T \mathbf{p} \cdot \delta\mathbf{v} dt \text{ en } (0, T). \quad (4.11)$$

Integrando por partes, y tomando en cuenta la simetría de las matrices Γ y K , de (4.10), (4.11) se sigue que

$$\Gamma \mathbf{p}(T) \cdot \delta\mathbf{y}(T) + \int_0^T -\Gamma \frac{d\mathbf{p}}{dt} \cdot \delta\mathbf{y} dt + \int_0^T (K + C) \mathbf{p} \cdot \delta\mathbf{y} dt = \int_0^T \mathbf{p} \cdot \delta\mathbf{v} dt \text{ en } (0, T).$$

Así que

$$\begin{aligned} & \Gamma \mathbf{p}(T) \cdot \delta \mathbf{y}(T) + \int_0^T \left\{ -\Gamma \frac{d}{dt} \mathbf{p} + (K + C) \mathbf{p} \right\} \cdot \delta \mathbf{y} dt \\ &= \int_0^T \mathbf{p} \cdot \delta \mathbf{v} dt \text{ en } (0, T), \end{aligned} \quad (4.12)$$

Supóngase que la función vectorial \mathbf{p} es solución del siguiente sistema (**adjunto**):

$$\begin{cases} -\Gamma \frac{d\mathbf{p}}{dt} + (K + C) \mathbf{p} = k_1 \mathbf{y}, \\ \Gamma \mathbf{p}(T) = k_2 \mathbf{y}(T). \end{cases} \quad (4.13)$$

Se sigue de (4.12), (4.13) que

$$k_2 \mathbf{y}(T) \cdot \delta \mathbf{y}(T) + k_1 \int_0^T \mathbf{y} \cdot \delta \mathbf{y} dt = \int_0^T \mathbf{p} \cdot \delta \mathbf{v} dt. \quad (4.14)$$

Combinando (4.8) y (4.14) obtenemos

$$\int_0^T DJ(\mathbf{v}) \cdot \delta \mathbf{v} dt = \int_0^T \mathbf{v} \cdot \delta \mathbf{v} dt + k_1 \int_0^T \mathbf{y} \cdot \delta \mathbf{y} dt + k_2 \mathbf{y}(T) \cdot \delta \mathbf{y}(T) = \int_0^T (\mathbf{v} + \mathbf{p}) \cdot \delta \mathbf{v} dt, \quad (4.15)$$

lo que implica a su vez que

$$DJ(\mathbf{v}) = \mathbf{v} + \mathbf{p}. \quad (4.16)$$

Nota 4.1. Hasta ahora hemos supuesto que en cada junta i está actuando un control v_i pero el método usado para calcular DJ también puede aplicarse si uno considera controlar la transición (desde $\mathbf{y}(0) = \delta\phi(0) = \delta\theta$ hasta $\mathbf{y}(T) = \delta\phi(T) = \mathbf{0}$) usando sólo uno o dos controles.

4.3.3 Condiciones de optimalidad para el problema (4.5)

Sea \mathbf{u} la solución del problema (4.5) y denotemos por \mathbf{y} (respectivamente, \mathbf{p}) la solución correspondiente al sistema de estado (4.3) (respectivamente, del sistema adjunto (4.13)). De acuerdo a la subsección 4.3.2 $DJ(\mathbf{u}) = \mathbf{0}$ es equivalente al siguiente sistema de optimalidad:

$$\mathbf{u} = -\mathbf{p}. \quad (4.17)$$

$$\gamma_1 \frac{dy_1}{dt} + \kappa_1(y_1 - y_2) + C_1 y_1 = u_1, \text{ en } (0, T), \quad (4.18)$$

$$\gamma_2 \frac{dy_2}{dt} + \kappa_1(y_2 - y_1) + \kappa_2(y_2 - y_3) + C_2 y_2 = u_2, \text{ en } (0, T), \quad (4.19)$$

$$\gamma_3 \frac{dy_3}{dt} + \kappa_2(y_3 - y_2) + C_3 y_3 = u_3, \text{ en } (0, T), \quad (4.20)$$

$$\mathbf{y}(0) = \delta\theta, \quad (4.21)$$

$$-\Gamma \frac{d\mathbf{p}}{dt} + (K + C) \mathbf{p} = k_1 \mathbf{y}, \text{ en } (0, T), \quad (4.22)$$

$$\Gamma \mathbf{p}(T) = k_2 \mathbf{y}(T). \quad (4.23)$$

A la inversa, se puede mostrar (ver, e.g., [37]) que el sistema (4.17)–(4.23) caracteriza \mathbf{u} como la solución (necesariamente única en este caso) del problema de control (4.5). Las condiciones de optimalidad (4.17)–(4.23) juegan un papel crucial en lo que respecta a la solución iterativa del problema de control (4.5).

4.3.4 Ecuación funcional satisfecha por el control óptimo

Nuestra meta en este apartado es mostrar que la condición de optimalidad $DJ(\mathbf{u}) = \mathbf{0}$ se puede escribir también como

$$\mathbf{A}\mathbf{u} = \beta, \quad (4.24)$$

donde el operador lineal \mathbf{A} es un isomorfismo simétrico fuertemente elíptico de \mathcal{U} en si mismo (un automorfismo \mathcal{U}) y $\beta \in \mathcal{U}$. Un candidato para \mathbf{A} es el operador lineal de \mathcal{U} en si mismo definido como

$$\mathbf{A}\mathbf{v} = \mathbf{v} + \mathbf{p}(\mathbf{v}),$$

con $\mathbf{p}(\mathbf{v})$ obtenido a partir de \mathbf{v} via la solución sucesiva de los dos problemas siguientes:

$$\begin{cases} \Gamma \frac{d}{dt} \mathbf{y} + (K + C)\mathbf{y} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \text{ en } (0, T), \\ \mathbf{y}(0) = \mathbf{0}. \end{cases} \quad (4.25)$$

el cual es un problema hacia adelante en el tiempo, y

$$\begin{cases} -\Gamma \frac{d}{dt} \mathbf{p} + (K + C)\mathbf{p} = k_1 \mathbf{y}, \\ \Gamma \mathbf{p}(T) = k_2 \mathbf{y}(T). \end{cases} \quad (4.26)$$

el cual es un problema hacia atrás en el tiempo. Para ver que el operador \mathbf{A} es un isomorfismo simétrico y fuertemente elíptico de \mathcal{U} en si mismo consideramos $\mathbf{v}^1, \mathbf{v}^2$ en \mathcal{U} y definimos $\mathbf{y}^i, \mathbf{p}^i$ por

$$\mathbf{y}^i = \mathbf{y}(\mathbf{v}^i), \quad \mathbf{p}^i = \mathbf{p}(\mathbf{v}^i), \quad i = 1, 2;$$

tenemos entonces

$$\int_0^T (A\mathbf{v}^1) \cdot \mathbf{v}^2 dt = \int_0^T (\mathbf{v}^1 + \mathbf{p}^1) \cdot \mathbf{v}^2 dt = \int_0^T \mathbf{v}^1 \cdot \mathbf{v}^2 dt + \int_0^T \mathbf{p}^1 \cdot \mathbf{v}^2 dt. \quad (4.27)$$

Por otro lado, tenemos, empezando con la ecuación diferencial en (4.25) y usando integración por partes, que

$$\begin{aligned} 0 &= \int_0^T \left(\Gamma \frac{d}{dt} \mathbf{y}^2 + (K + C)\mathbf{y}^2 - \mathbf{v}^2 \right) \cdot \mathbf{p}^1 dt \\ &= \Gamma \mathbf{p}^1(T) \cdot \mathbf{y}^2(T) + \int_0^T \left(-\Gamma \frac{d}{dt} \mathbf{p}^1 + (K + C)\mathbf{p}^1 \right) \cdot \mathbf{y}^2 dt - \int_0^T \mathbf{v}^2 \cdot \mathbf{p}^1 dt \\ &= k_2 \mathbf{y}^1(T) \cdot \mathbf{y}^2(T) + \int_0^T k_1 \mathbf{y}^1 \cdot \mathbf{y}^2 dt - \int_0^T \mathbf{v}^2 \cdot \mathbf{p}^1 dt. \end{aligned}$$

Esto implica que

$$\int_0^T \mathbf{v}^2 \cdot \mathbf{p}^1 dt. = k_2 \mathbf{y}^1(T) \cdot \mathbf{y}^2(T) + \int_0^T k_1 \mathbf{y}^1 \cdot \mathbf{y}^2 dt. \quad (4.28)$$

Combinando (4.27) con (4.28) obtenemos

$$\int_0^T (\mathbf{A}\mathbf{v}^1) \cdot \mathbf{v}^2 dt = \int_0^T \mathbf{v}^1 \cdot \mathbf{v}^2 dt + k_2 \mathbf{y}^1(T) \cdot \mathbf{y}^2(T) + \int_0^T k_1 \mathbf{y}^1 \cdot \mathbf{y}^2 dt. \quad (4.29)$$

La relación (4.29) implica la simetría de \mathbf{A} ; también tenemos que

$$\int_0^T (\mathbf{A}\mathbf{v}) \cdot \mathbf{v} dt \geq \int_0^T |\mathbf{v}|^2 dt, \quad \forall \mathbf{v} \in \mathcal{U},$$

lo cual implica la elipticidad fuerte de \mathbf{A} sobre \mathcal{U} (también es continuo). El operador lineal \mathbf{A} , al ser continuo y fuertemente elíptico sobre \mathcal{U} , es un automorfismo de \mathcal{U} . Para identificar el lado derecho β de la ecuación (4.24), introducimos \mathbf{Y}_0 y \mathbf{P}_0 definido como la solución de

$$\begin{cases} \Gamma \frac{d}{dt} \mathbf{Y}_0 + (K + C) \mathbf{Y}_0 = \mathbf{0} \text{ en } (0, T), \\ \mathbf{Y}_0(0) = \delta\theta, \end{cases} \quad (4.30)$$

y

$$\begin{cases} -\Gamma \frac{d}{dt} \mathbf{P}_0 + (K + C) \mathbf{P}_0 = k_1 \mathbf{Y}_0, \\ \Gamma \mathbf{P}_0(T) = k_2 \mathbf{Y}_0(T). \end{cases} \quad (4.31)$$

Suponemos ahora que \mathbf{y} y \mathbf{p} satisface las condiciones de optimalidad. Definimos

$$\begin{aligned} \bar{\mathbf{y}} &= \mathbf{y} - \mathbf{Y}_0, \\ \bar{\mathbf{p}} &= \mathbf{p} - \mathbf{P}_0. \end{aligned}$$

Restando (4.30) y (4.31) de (4.25) y (4.26) obtenemos

$$\begin{cases} \Gamma \frac{d}{dt} \bar{\mathbf{y}} + (K + C) \bar{\mathbf{y}} = \mathbf{u} \text{ en } (0, T), \\ \bar{\mathbf{y}}(0) = \mathbf{0}, \end{cases} \quad (4.32)$$

y

$$\begin{cases} -\Gamma \frac{d}{dt} \bar{\mathbf{p}} + (K + C) \bar{\mathbf{p}} = k_1 \bar{\mathbf{y}}, \\ \Gamma \bar{\mathbf{p}}(T) = k_2 \bar{\mathbf{y}}(T). \end{cases} \quad (4.33)$$

Por la definición del operador \mathbf{A} se sigue que

$$\mathbf{A}\mathbf{u} = \mathbf{u} + \bar{\mathbf{p}} = -\mathbf{p} + \bar{\mathbf{p}} = -\mathbf{P}_0; \quad (4.34)$$

el lado derecho de (4.34) es el vector β que estábamos buscando.

Por las propiedades de \mathbf{A} , se puede resolver el problema (4.34) usando un algoritmo de gradiente conjugado operando en un espacio de Hilbert \mathcal{U} . Este algoritmo se describirá en la siguiente subsección.

4.3.5 Solución por gradiente conjugado del problema de control (4.5)

Generalidades

El problema (4.34) se puede escribir en forma variacional como

$$\begin{aligned} \mathbf{u} &\in \mathcal{U} (= L^2(0, T; 3)), \\ \int_0^T (\mathbf{A}\mathbf{u}) \cdot \mathbf{v} \, dt &= - \int_0^T \mathbf{P}_0 \cdot \mathbf{v} \, dt, \forall \mathbf{v} \in \mathcal{U}. \end{aligned} \quad (4.35)$$

Por la simetría y \mathcal{U} -elipticidad de la forma bilineal

$$\{\mathbf{v}, \mathbf{w}\} \rightarrow \int_0^T (\mathbf{A}\mathbf{v}) \cdot \mathbf{w} \, dt,$$

el problema variacional (4.35) es un caso particular de la clase de problemas variacionales lineales descritos en el Apéndice A.

Sobre la solución del problema de minimización (4.5) cuando $\mathcal{U} = L^2(0, T; 3)$

Recordemos que $\mathcal{U} = L^2(0, T; 3)$ es un espacio de Hilbert para el producto interior $\{\mathbf{v}, \mathbf{w}\} \rightarrow \int_0^T \mathbf{v} \cdot \mathbf{w} \, dt$ y la norma asociada $\mathbf{v} \rightarrow \sqrt{\int_0^T |\mathbf{v}|^2 \, dt}$, implicando que el problema (4.5)-(4.35) se puede resolver aplicando el algoritmo de gradiente conjugado (A.2)-(A.8). Para nuestro caso ese algoritmo toma la siguiente forma:

- Sea

$$\mathbf{u}^0 \text{ dado en } L^2(0, T; 3) \text{ (} \mathbf{u}^0 = \mathbf{0} \text{ por ejemplo)}. \quad (4.36)$$

- Resolver para \mathbf{y}^0

$$\begin{cases} \Gamma \frac{d}{dt} \mathbf{y}^0 + (K + C) \mathbf{y}^0 = \begin{pmatrix} u_1^0 \\ u_2^0 \\ u_3^0 \end{pmatrix} \text{ en } (0, T), \\ \mathbf{y}^0(0) = \delta \theta. \end{cases} \quad (4.37)$$

y enseguida resolver para \mathbf{p}^0

$$\begin{cases} -\Gamma \frac{d}{dt} \mathbf{p}^0 + (K + C) \mathbf{p}^0 = k_1 \mathbf{y}^0, \text{ en } (0, T), \\ \Gamma \mathbf{p}^0(T) = k_2 \mathbf{y}^0(T). \end{cases} \quad (4.38)$$

- Tomar

$$\mathbf{g}^0 = \mathbf{u}^0 + \mathbf{p}^0. \quad (4.39)$$

- Si $\int_0^T |\mathbf{g}^0|^2 \, dt \leq tol^2 \max[1, \int_0^T |\mathbf{u}^0|^2 \, dt]$, tomar $\mathbf{u} = \mathbf{u}^0$; en otro caso, hacer

$$\mathbf{w}^0 = \mathbf{g}^0. \quad (4.40)$$

Para $q \geq 0$, \mathbf{u}^q , \mathbf{g}^q y \mathbf{w}^q conocidos, calcular \mathbf{u}^{q+1} , \mathbf{g}^{q+1} y si es necesario, calcular \mathbf{w}^{q+1} como sigue:

- Resolver para $\bar{\mathbf{y}}^q$

$$\begin{cases} \Gamma \frac{d}{dt} \bar{\mathbf{y}}^q + (K + C) \bar{\mathbf{y}}^q = \begin{pmatrix} w_1^q \\ w_2^q \\ w_3^q \end{pmatrix} \text{ en } (0, T), \\ \mathbf{y}^0(0) = \mathbf{0}. \end{cases} \quad (4.41)$$

y entonces resolver para $\bar{\mathbf{p}}^q$

$$\begin{cases} -\Gamma \frac{d}{dt} \bar{\mathbf{p}}^q + (K + C) \bar{\mathbf{p}}^q = k_1 \bar{\mathbf{y}}^q, \text{ en } (0, T), \\ \Gamma \bar{\mathbf{p}}^q(T) = k_2 \bar{\mathbf{y}}^q(T). \end{cases} \quad (4.42)$$

Tomar

$$\bar{\mathbf{g}}^q = \mathbf{w}^q + \bar{\mathbf{p}}^q \quad (4.43)$$

y

$$\rho_q = \|\mathbf{g}^q\|^2 / \int_0^T \bar{\mathbf{g}}^q \cdot \mathbf{w}^q dt. \quad (4.44)$$

- Hacer

$$\mathbf{u}^{q+1} = \mathbf{u}^q - \rho_q \mathbf{w}^q. \quad (4.45)$$

- Tomar

$$\mathbf{g}^{q+1} = \mathbf{g}^q - \rho_q \bar{\mathbf{g}}^q. \quad (4.46)$$

- Si $\frac{\int_0^T |\mathbf{g}^{q+1}|^2 dt}{\max[\int_0^T |\mathbf{g}^0|^2 dt, \int_0^T |\mathbf{u}^{q+1}|^2 dt]} \leq tol^2$, tomar $\mathbf{u} = \mathbf{u}^{q+1}$; en otro caso, calcular

$$\gamma_q = \frac{\int_0^T |\mathbf{g}^{q+1}|^2 dt}{\int_0^T |\mathbf{g}^q|^2 dt} \quad (4.47)$$

y hacer

$$\mathbf{w}^{q+1} = \mathbf{g}^{q+1} + \gamma_q \mathbf{w}^q. \quad (4.48)$$

- Hacer $q + 1 \rightarrow q$ y regresar a (4.41) .
- Fin del algoritmo.

La variable de estado \mathbf{y}^q puede ser actualizada simultáneamente con el control \mathbf{u}^q ya que para todo q tenemos

$$\begin{cases} \Gamma \frac{d}{dt} \mathbf{y}^q + (K + C) \mathbf{y}^q = \mathbf{u}^q, \text{ en } (0, T), \\ \mathbf{y}^q(0) = \delta\theta. \end{cases} \quad (4.49)$$

Así que,

$$\begin{cases} \Gamma \frac{d}{dt} (\mathbf{y}^{q+1} - \mathbf{y}^q) + (K + C) (\mathbf{y}^{q+1} - \mathbf{y}^q) = \mathbf{u}^{q+1} - \mathbf{u}^q = -\rho_q \mathbf{w}^q, \text{ en } (0, T), \\ (\mathbf{y}^{q+1} - \mathbf{y}^q)(0) = \mathbf{0}, \end{cases} \quad (4.50)$$

lo cual, por definición de $\bar{\mathbf{y}}^q$, implica que

$$\mathbf{y}^{q+1} - \mathbf{y}^q = -\rho_q \bar{\mathbf{y}}^q \implies \mathbf{y}^{q+1} = \mathbf{y}^q - \rho_q \bar{\mathbf{y}}^q.$$

La implementación práctica del algoritmo (4.36)-(4.48), usando una discretización por diferencias finitas del problema (4.5), se discutirá en la siguiente sección.

4.4 Implementación práctica del algoritmo (4.36)-(4.48) via una aproximación de diferencias finitas del problema (4.5)

4.4.1 Aproximación de diferencias finitas del problema (4.5) cuando $\mathcal{U} = L^2(0, T; 3)$

Aproximamos (4.5) cuando $\mathcal{U} = (L^2(0, T))^3$ por

$$\begin{cases} \mathbf{u}^{\Delta t} \in \mathcal{U}^{\Delta t}, \\ J^{\Delta t}(\mathbf{u}^{\Delta t}) \leq J^{\Delta t}(\mathbf{v}), \forall \mathbf{v} \in \mathcal{U}^{\Delta t}, \end{cases} \quad (4.51)$$

donde:

- $\Delta t = T/N$ con N un entero positivo “grande”.
- $\mathcal{U}^{\Delta t} = (\mathbb{R}^3)^N$.

El funcional de costo $J^{\Delta t}$ se define por

$$J^{\Delta t}(\mathbf{v}) = \frac{\Delta t}{2} \sum_{n=1}^N |\mathbf{v}^n|^2 + \frac{k_1 \Delta t}{2} \sum_{n=1}^N |\mathbf{y}^n|^2 + \frac{k_2}{2} \|\mathbf{y}^N\|^2,$$

con $\mathbf{v} = \{\mathbf{v}^n\}_{n=1}^N$ y $\{\mathbf{y}^n\}_{n=1}^N$ obtenido de \mathbf{v} y $\delta\theta$ via la siguiente variante discreta de (4.4):

$$\mathbf{y}^0 = \delta\theta, \quad (4.52)$$

y para $n = 1, \dots, N$

$$\left\{ \Gamma \frac{\mathbf{y}^n - \mathbf{y}^{n-1}}{\Delta t} + (K + C)\mathbf{y}^n = \begin{pmatrix} v_1^n \\ v_2^n \\ v_3^n \end{pmatrix} \text{ en } (0, T), \right. \quad (4.53)$$

Para calcular \mathbf{y}^n tenemos que resolver un sistema lineal del tipo

$$(\Gamma + \Delta t(K + C))\mathbf{y}^n = RHS^n. \quad (4.54)$$

La matriz $\Gamma + \Delta t(K + C)$ de 3×3 es simétrica y definida positiva, así que resolver (4.54) es fácil.

4.4.2 Condiciones de Optimalidad y solución por gradiente conjugado de (4.51)

Cálculo de $DJ^{\Delta t}(\mathbf{v})$

Puesto que deseamos usar el algoritmo de gradiente conjugado (A.2)-(A.8) para resolver el problema discreto (4.51), calculamos primero $DJ^{\Delta t}(\mathbf{v})$. En $\mathcal{U}^{\Delta t} = (\mathbb{R}^3)^N$ usaremos el siguiente producto interior

$$(\mathbf{v}, \mathbf{w})_{\Delta t} = \Delta t \sum_{n=1}^N \mathbf{v}^n \cdot \mathbf{w}^n, \quad \forall \mathbf{v} = \{\mathbf{v}^n\}_{n=1}^N, \quad \mathbf{w} = \{\mathbf{w}^n\}_{n=1}^N \in (\mathbb{R}^3)^N.$$

Tenemos que

$$\delta J^{\Delta t}(\mathbf{v}) = \Delta t \sum_{n=1}^N \mathbf{v}^n \cdot \delta \mathbf{v}^n + k_1 \Delta t \sum_{n=1}^N \mathbf{y}^n \cdot \delta \mathbf{y}^n + k_2 \mathbf{y}^N \cdot \delta \mathbf{y}^N, \quad (4.55)$$

donde $\{\delta \mathbf{y}^n\}_{n=0}^N$ se obtiene por perturbación de (4.52), (4.53), esto es

$$\delta \mathbf{y}^0 = \mathbf{0}, \quad (4.56)$$

y para $n = 1, \dots, N$

$$\Gamma \frac{\delta \mathbf{y}^n - \delta \mathbf{y}^{n-1}}{\Delta t} + (K + C) \delta \mathbf{y}^n = \begin{pmatrix} \delta v_1^n \\ \delta v_2^n \\ \delta v_3^n \end{pmatrix}. \quad (4.57)$$

Introducimos $\{\mathbf{p}^n\}_{n=1}^N \in (\mathbb{R}^3)^N$. Tomando el producto interior de \mathbf{p}^n con cada lado de la ecuación (4.57), obtenemos, después de hacer la sumatoria y la multiplicación por Δt :

$$\begin{aligned} & \Delta t \sum_{n=1}^N \Gamma \frac{\delta \mathbf{y}^n - \delta \mathbf{y}^{n-1}}{\Delta t} \cdot \mathbf{p}^n + \Delta t \sum_{n=1}^N (K + C) \delta \mathbf{y}^n \cdot \mathbf{p}^n \\ &= \Delta t \sum_{n=1}^N \delta \mathbf{v}^n \cdot \mathbf{p}^n. \end{aligned} \quad (4.58)$$

Aplicando sumación por partes (versión discreta de la integración por partes) a la primera sumatoria en la relación (4.58), y considerando que $\delta \mathbf{y}^0 = \mathbf{0}$, obtenemos:

$$\begin{aligned} & \Gamma \mathbf{p}^{N+1} \cdot \delta \mathbf{y}^N + \Delta t \sum_{n=1}^N \Gamma \frac{\mathbf{p}^n - \mathbf{p}^{n+1}}{\Delta t} \cdot \delta \mathbf{y}^n + \Delta t \sum_{n=1}^N (K + C) \mathbf{p}^n \cdot \delta \mathbf{y}^n \\ &= \Delta t \sum_{n=1}^N \delta \mathbf{v}^n \cdot \mathbf{p}^n. \end{aligned} \quad (4.59)$$

o equivalentemente

$$\begin{aligned} & \Gamma \mathbf{p}^{N+1} \cdot \delta \mathbf{y}^N + \Delta t \sum_{n=1}^N \left\{ \Gamma \frac{\mathbf{p}^n - \mathbf{p}^{n+1}}{\Delta t} + (K + C) \mathbf{p}^n \right\} \cdot \delta \mathbf{y}^n \\ &= \Delta t \sum_{n=1}^N \delta \mathbf{v}^n \cdot \mathbf{p}^n. \end{aligned} \quad (4.60)$$

Suponemos ahora que $\{\mathbf{p}^n\}_{n=1}^{N+1}$ verifica el siguiente sistema adjunto discreto:

$$\Gamma \mathbf{p}^{N+1} = k_2 \mathbf{y}^N, \quad (4.61)$$

y para $n = N, \dots, 1$

$$\Gamma \frac{\mathbf{p}^n - \mathbf{p}^{n+1}}{\Delta t} + (K + C) \mathbf{p}^n = k_1 \mathbf{y}^n. \quad (4.62)$$

Se sigue de (4.55), (4.60)-(4.62) que

$$\delta J^{\Delta t}(\mathbf{v}) = \Delta t \sum_{n=1}^N \mathbf{v}^n \cdot \delta \mathbf{v}^n + k_1 \Delta t \sum_{n=1}^N \mathbf{y}^n \cdot \delta \mathbf{y}^n + k_2 \mathbf{y}^N \cdot \delta \mathbf{y}^N = \Delta t \sum_{n=1}^N (\mathbf{v}^n + \mathbf{p}^n) \cdot \delta \mathbf{v}^n, \quad (4.63)$$

esto es

$$DJ^{\Delta t}(\mathbf{v}) = \{\mathbf{v}^n + \mathbf{p}^n\}_{n=1}^N. \quad (4.64)$$

Condiciones de optimalidad para (4.51)

Las condiciones de optimalidad para el problema discreto (4.51) son

$$\mathbf{u}^n = -\mathbf{p}^n, \quad n = 1, \dots, N, \quad (4.65)$$

$$\mathbf{y}^0 = \delta \theta, \quad (4.66)$$

$$\Gamma \frac{\mathbf{y}^n - \mathbf{y}^{n-1}}{\Delta t} + (K + C) \mathbf{y}^n = \begin{pmatrix} u_1^n \\ u_2^n \\ u_3^n \end{pmatrix} \text{ en } (0, T), \quad n = 1, \dots, N, \quad (4.67)$$

$$\Gamma \mathbf{p}^{N+1} = k_2 \mathbf{y}^N, \quad (4.68)$$

$$\Gamma \frac{\mathbf{p}^n - \mathbf{p}^{n+1}}{\Delta t} + (K + C) \mathbf{p}^n = k_1 \mathbf{y}^n, \quad n = N, \dots, 1 \quad (4.69)$$

Ecuación funcional para la solución discreta del problema de control (4.51)

Siguiendo el esquema del caso continuo podemos mostrar que la versión discreta $\mathbf{A}^{\Delta t}$ del operador \mathbf{A} y la versión discreta $\beta^{\Delta t}$ de β satisface la ecuación

$$\mathbf{A}^{\Delta t} \mathbf{u}^{\Delta t} = \beta^{\Delta t}, \quad (4.70)$$

donde $\mathbf{u}^{\Delta t}$ es el control discreto satisfaciendo la condición de optimalidad (4.65). El operador $\mathbf{A}^{\Delta t}$ tiene las mismas propiedades que la versión continua: simétrico, fuertemente elíptico y continuo, permitiéndonos usar un algoritmo de gradiente conjugado como (A.2)-(A.8) para resolver (4.70).

Solución por gradiente conjugado del problema discreto de control (4.51)

Usando $\mathbf{y}_q^n = \{\mathbf{y}_{iq}^n\}_{i=1}^3$ para denotar el valor discreto de la función vector-valuada \mathbf{y} en el tiempo $n\Delta t$ e iteración q ; similarmente, \mathbf{u}_q^n denotará el valor discreto del control \mathbf{u} en el tiempo $n\Delta t$ e iteración q . El algoritmo de gradiente conjugado (A.2)-(A.8) para resolver el problema de dimensión finita (4.51) queda como:

- Suponer que

$$\mathbf{u}_0 = \{\{u_{i0}^n\}_{i=1}^3\}_{n=1}^N \text{ se da en } \mathcal{U}_{ad}^{\Delta t} = (\mathbb{R}^3)^N \text{ (}\mathbf{u}_0 = \mathbf{0} \text{ por ejemplo)}. \quad (4.71)$$

- Calcular $\{\mathbf{y}_0^n\}_{n=0}^N = \{\{\mathbf{y}_{i0}^n\}_{i=1}^3\}_{n=0}^N$ y $\{\mathbf{p}_0^n\}_{n=1}^{N+1} = \{\{\mathbf{p}_{i0}^n\}_{i=1}^3\}_{n=1}^{N+1}$, resolviendo las siguientes dos ecuaciones en este orden: primero

$$\left\{ \begin{array}{l} \mathbf{y}_0^0 = \delta\theta, \\ \text{para } n = 1, \dots, N \text{ resolver} \\ \Gamma \frac{\mathbf{y}_0^n - \mathbf{y}_0^{n-1}}{\Delta t} + (K + C)\mathbf{y}_0^n = \begin{pmatrix} u_{10}^n \\ u_{20}^n \\ u_{30}^n \end{pmatrix}, \end{array} \right. \quad (4.72)$$

y a continuación

$$\left\{ \begin{array}{l} \Gamma \mathbf{p}_0^{N+1} = k_2 \mathbf{y}_0^N, \\ \text{para } n = N, \dots, 1 \text{ resolver} \\ \Gamma \frac{\mathbf{p}_0^n - \mathbf{p}_0^{n+1}}{\Delta t} + (K + C)\mathbf{p}_0^n = k_1 \mathbf{y}_0^n. \end{array} \right. \quad (4.73)$$

- Hacer

$$\mathbf{g}_0 = \{\mathbf{g}_0^n\}_{n=1}^N = \{\mathbf{u}_0^n + \mathbf{p}_0^n\}_{n=1}^N. \quad (4.74)$$

- Si

$$\frac{\Delta t \sum_{n=1}^N |\mathbf{g}_0^n|^2}{\max[1, \Delta t \sum_{n=1}^N |\mathbf{u}_0^n|^2]} \leq tol^2, \text{ con } |\mathbf{g}_0^n|^2 = |g_{10}^n|^2 + |g_{20}^n|^2 + |g_{30}^n|^2,$$

tomar $\mathbf{u}^{\Delta t} = \mathbf{u}_0$; en otro caso, hacer

$$\mathbf{w}_0 = \mathbf{g}_0. \quad (4.75)$$

Para $q \geq 0$, \mathbf{u}_q , \mathbf{g}_q y \mathbf{w}_q conocidos, los dos últimos diferentes de $\mathbf{0}$, calcular \mathbf{u}_{q+1} , \mathbf{g}_{q+1} y, de ser necesario \mathbf{w}_{q+1} como sigue:

- Resolver

$$\left\{ \begin{array}{l} \bar{\mathbf{y}}_q^0 = \mathbf{0}, \\ \text{para } n = 1, \dots, N \text{ resolver} \\ \Gamma \frac{\bar{\mathbf{y}}_q^n - \bar{\mathbf{y}}_q^{n-1}}{\Delta t} + (K + C)\bar{\mathbf{y}}_q^n = \begin{pmatrix} w_{1q}^n \\ w_{2q}^n \\ w_{3q}^n \end{pmatrix}, \end{array} \right. \quad (4.76)$$

y entonces

$$\left\{ \begin{array}{l} \Gamma \bar{\mathbf{p}}_q^{N+1} = k_2 \bar{\mathbf{y}}_q^N, \\ \text{para } n = N, \dots, 1 \text{ resolver} \\ \Gamma \frac{\bar{\mathbf{p}}_q^n - \bar{\mathbf{p}}_q^{n+1}}{\Delta t} + (K + C)\bar{\mathbf{p}}_q^n = k_1 \bar{\mathbf{y}}_q^n. \end{array} \right. \quad (4.77)$$

Hacer

$$\bar{\mathbf{g}}_q = \mathbf{w}_q + \bar{\mathbf{p}}_q \quad (4.78)$$

y

$$\rho_q = \Delta t \sum_{n=1}^N |\mathbf{g}_q^n|^2 / (\Delta t \sum_{n=1}^N \bar{\mathbf{g}}_q^n \cdot \mathbf{w}_q^n). \quad (4.79)$$

- Calcular

$$\mathbf{u}_{q+1} = \mathbf{u}_q - \rho_q \mathbf{w}_q. \quad (4.80)$$

y

$$\mathbf{g}_{q+1} = \{\mathbf{g}_{q+1}^n\}_{n=1}^N = \mathbf{g}_q - \rho_q \bar{\mathbf{g}}_q. \quad (4.81)$$

- Si

$$\frac{\Delta t \sum_{n=1}^N |\mathbf{g}_{q+1}^n|^2}{\max[\Delta t \sum_{n=1}^N |\mathbf{g}_0^n|^2, \Delta t \sum_{n=1}^N |\mathbf{u}_{q+1}^n|^2]} \leq tol^2,$$

tomar $\mathbf{u}^{\Delta t} = \mathbf{u}_{q+1}$; en otro caso, calcular

$$\gamma_q = \frac{\sum_{n=1}^N |\mathbf{g}_{q+1}^n|^2}{\sum_{n=1}^N |\mathbf{g}_q^n|^2} \quad (4.82)$$

y hacer

$$\mathbf{w}_{q+1} = \mathbf{g}_{q+1} + \gamma_q \mathbf{w}_q. \quad (4.83)$$

- Hacer $q + 1 \rightarrow q$ y regresar a (4.76) .
- Fin del algoritmo

Similar al caso continuo, podemos deducir que si $\mathbf{y}_q = \mathbf{y}(\mathbf{u}_q)$ entonces

$$\mathbf{y}_{q+1} - \mathbf{y}_q = -\rho_q \bar{\mathbf{y}}_q \implies \mathbf{y}_{q+1} = \mathbf{y}_q - \rho_q \bar{\mathbf{y}}_q.$$

4.5 Resultados Numéricos

En las secciones previas describimos una metodología y los algoritmos prácticos respectivos para encontrar un control actuando en cada junta con el fin de estabilizar la versión lineal perturbada del modelo AMJJ alrededor de un equilibrio inestable, esperando que estos controles estabilicen también el modelo no lineal original. Es fácil simplificar el procedimiento y obtener los respectivos algoritmos en el caso cuando deseamos controlar sólo con dos juntas (cualquier combinación de ellas) o bien con solo una junta. En esta sección resumimos los resultados de acuerdo al número de juntas sobre las que actúa un control. Los resultados numéricos nos mostrarán que es necesario llevar a cabo el proceso de control mediante al menos dos juntas para poder estabilizar el sistema alrededor de un equilibrio inestable. Puesto que el fenómeno que estamos considerando en este trabajo involucra un intervalo de tiempo largo (por cuestiones prácticas de $t = 0$ a $t = 40$ aproximadamente) y tomando ideas de la sección 2.11 de [34], donde se menciona que algunas

veces el control que estabiliza el modelo lineal podría no ser suficiente para estabilizar el sistema no lineal, el procedimiento para calcular los controles óptimos consiste en dividir el intervalo de tiempo en subintervalos de longitud 2 unidades, y calcular iterativamente los controles en cada subintervalo. Por esa razón presentamos, para el caso cuando se controla sólo mediante una o dos juntas, resultados donde se usa el subintervalo $[0, 2]$. En el caso final, cuando se intenta controlar mediante las tres juntas, se incluyen los resultados de controlar iterativamente el sistema en cada subintervalo hasta que se cubra el intervalo $[0, 20]$. Todos los resultados que se muestran a continuación se obtuvieron con $\Delta t = 1E - 2$.

4.5.1 Intentando controlar mediante una junta

Cuando controlamos con una sola junta, los resultados numéricos muestran que la función de perturbación \mathbf{y} no se aproxima a cero con el tiempo. Las figuras 4.5, 4.6 y 4.7 muestran el comportamiento de $\|\mathbf{y}^{\Delta t}(\cdot)\|$, $\mathbf{u}^{\Delta t}(\cdot)$ y $\ln\|\mathbf{g}_q^{\Delta t}\|$ respectivamente, para varios valores de k_1 y k_2 . Para estas figuras, el equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$ y se intenta controlar con la junta 1.

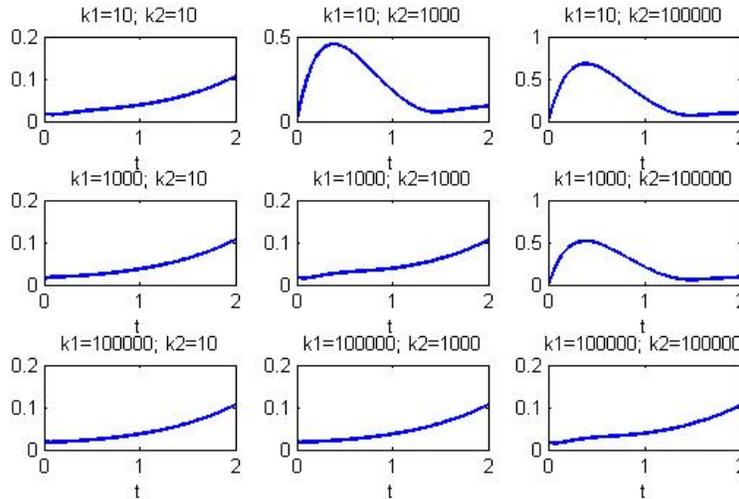


Figura 4.5: $\|\mathbf{y}^{\Delta t}(\cdot)\|$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se intenta controlar con la primera junta.

También se realizaron experimentos en donde se intentó controlar únicamente con la junta 2, así como con la junta 3. En ningún caso fue posible estabilizar el sistema alrededor del equilibrio inestable $\{1, 0, 0(u)\}$.

4.5.2 Intentando controlar con dos juntas

En las figuras 4.8, 4.9 y 4.10 se muestra el comportamiento de $\|\mathbf{y}^{\Delta t}(\cdot)\|$, $\mathbf{u}^{\Delta t}$ y $\ln\|\mathbf{g}_q^{\Delta t}\|$ respectivamente, para varios valores de k_1 y k_2 . Para estas figuras, el equilibrio inestable θ que se considera está dado por $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$ y las juntas a través de las cuales se intenta controlar son la 1 y la 2.

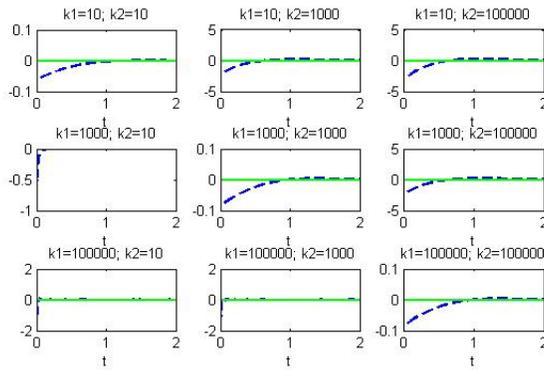


Figura 4.6: $\mathbf{u}^{\Delta t}$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se intenta controlar con la primera junta.

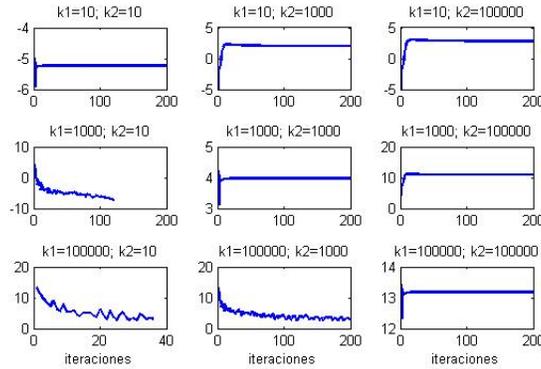


Figura 4.7: $\ln\|\mathbf{g}_q^{\Delta t}\|$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se intenta controlar con la primera junta.

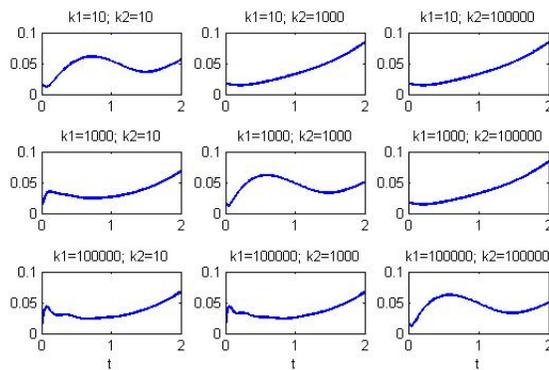


Figura 4.8: $\|\mathbf{y}^{\Delta t}(\cdot)\|$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las juntas 1 y 2.

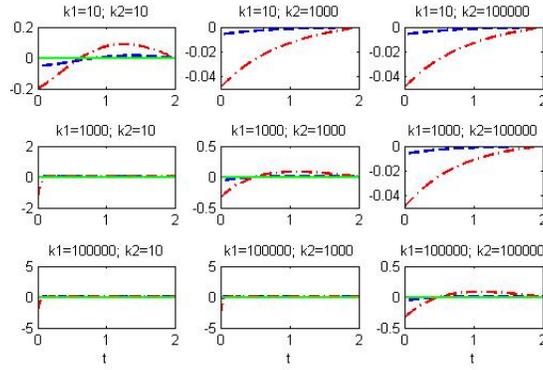


Figura 4.9: $\mathbf{u}^{\Delta t}$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las juntas 1 y 2.

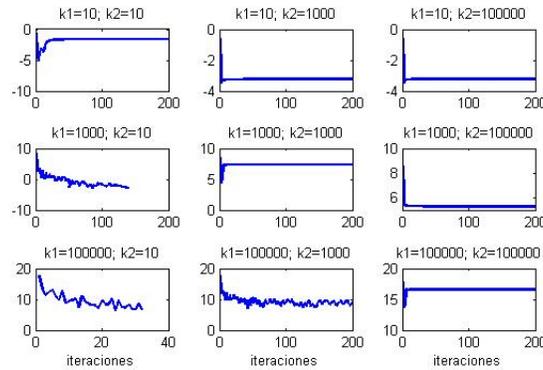


Figura 4.10: $\ln\|\mathbf{g}_q^{\Delta t}\|$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las juntas 1 y 2.

Está claro que no fue posible estabilizar el sistema alrededor de este equilibrio inestable. Por este motivo, se intentó estabilizar el sistema con las juntas 1 y 3, pero de nuevo no fue posible.

Por último, en las figuras 4.11, 4.12 y 4.13 se muestran los respectivos resultados cuando se controla a través de las juntas 2 y 3. Como se puede observar, en este caso se obtiene el mejor resultado (para todos los valores de k_1 y k_2).

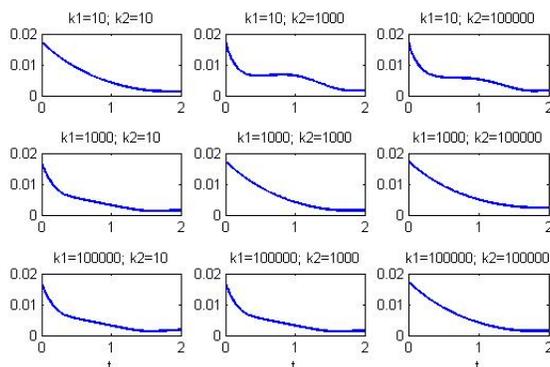


Figura 4.11: $\|\mathbf{y}^{\Delta t}(\cdot)\|$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las juntas 2 y 3.

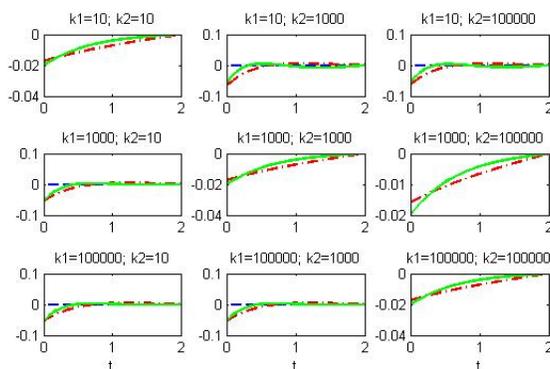


Figura 4.12: $\mathbf{u}^{\Delta t}$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las juntas 2 y 3.

4.5.3 Controlando mediante las tres juntas

En las figuras 4.14, 4.15 y 4.16 se muestran los resultados cuando se controla mediante las tres juntas. Como se puede ver, para todos los valores usados de los parámetros de penalización, se logra controlar el comportamiento del modelo lineal.

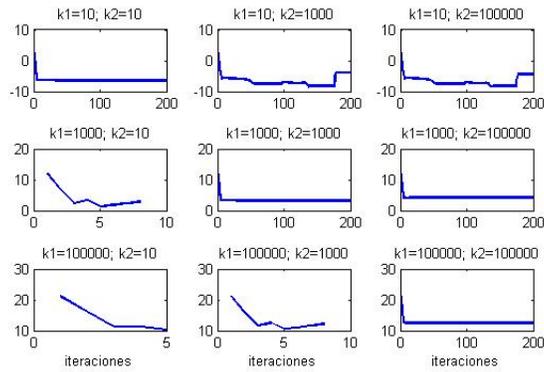


Figura 4.13: $\ln\|\mathbf{g}_q^{\Delta t}\|$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las juntas 2 y 3.

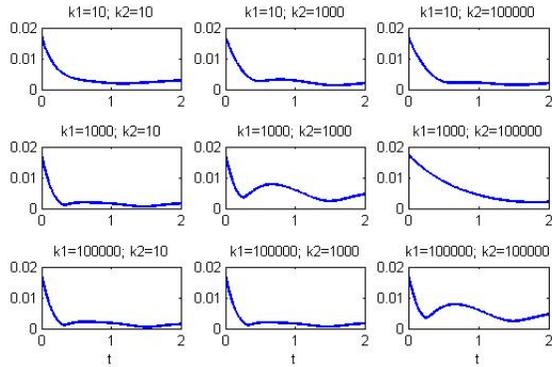


Figura 4.14: $\|\mathbf{y}^{\Delta t}(\cdot)\|$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las 3 juntas.

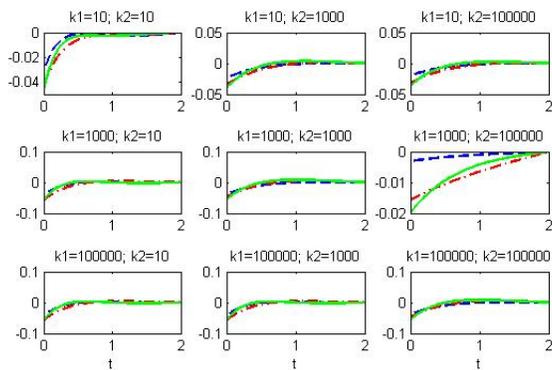


Figura 4.15: $\mathbf{u}^{\Delta t}$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las 3 juntas.

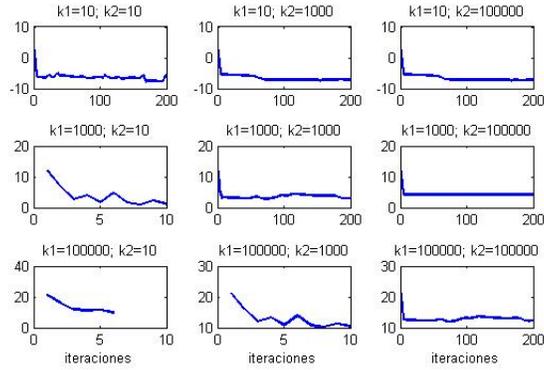


Figura 4.16: $\ln\|\mathbf{g}_q^{\Delta t}\|$ para varios valores de k_1 y k_2 . El equilibrio inestable θ es $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$. Se controla con las 3 juntas.

Recordemos que el propósito de utilizar el modelo lineal es calcular controles que sirvan también para estabilizar el modelo no lineal. Así que para el caso específico de los parámetros de penalización $k_1 = 1E + 3$ y $k_2 = 1E + 3$, en las figuras 4.17, 4.18 y 4.19 se muestran los controles, la norma de la solución del sistema lineal y la norma de la solución del sistema no lineal, respectivamente, cuando se controla con las tres juntas desde $t = 0$ hasta $t = 2$, considerando la perturbación inicial $\delta\theta = [1E - 2, 1E - 2, 1E - 2]$ al equilibrio inestable θ dado por $\{n_1, n_2, n_3\} = \{1, 0, 0(u)\}$.

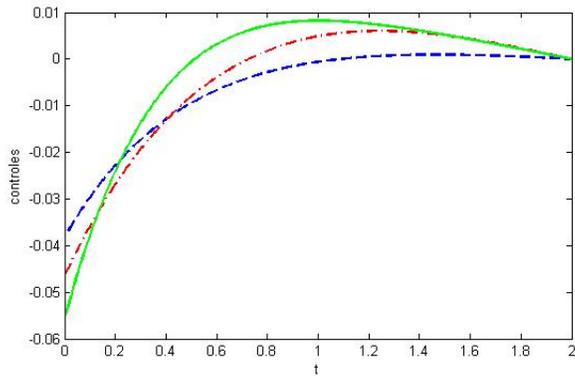


Figura 4.17: Controles para $T = 2$ cuando $k_1 = 1E + 3$ y $k_2 = 1E + 3$ y se controla con las tres juntas.

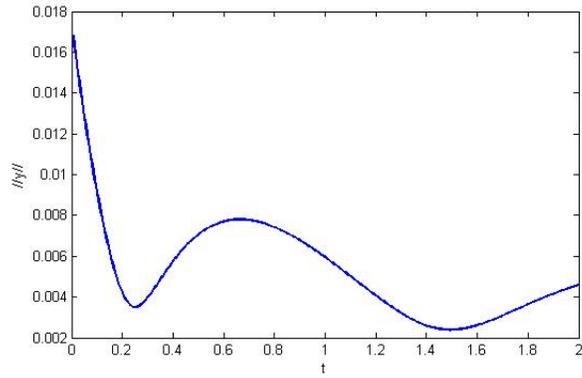


Figura 4.18: Norma euclídeana de la solución $\mathbf{y}^{\Delta t}$ del sistema lineal cuando se usan los controles de la figura 4.17.

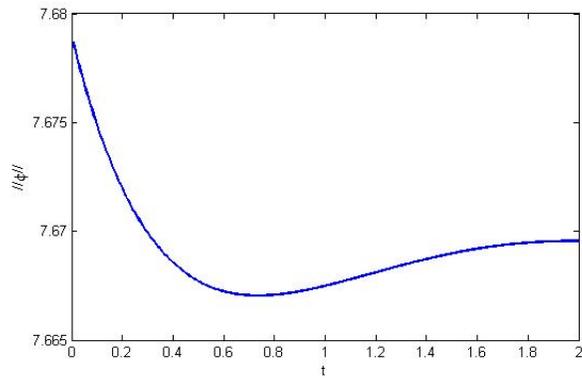


Figura 4.19: Norma euclídeana de la solución $\phi^{\Delta t}$ del sistema no lineal cuando se usan los controles de la figura 4.17.

En las figuras 4.20 y 4.21 se muestran las componentes de la solución para los modelos lineal y no lineal, respectivamente, cuando se aplican los controles de la figura 4.17.

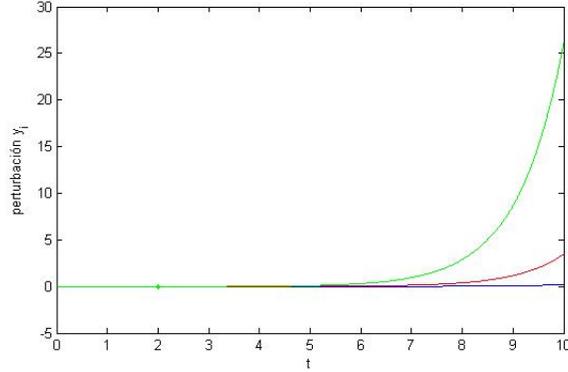


Figura 4.20: Componentes $y_i^{\Delta t}$ de la solución del modelo lineal perturbado, cuando se aplican los controles de la figura 4.17.

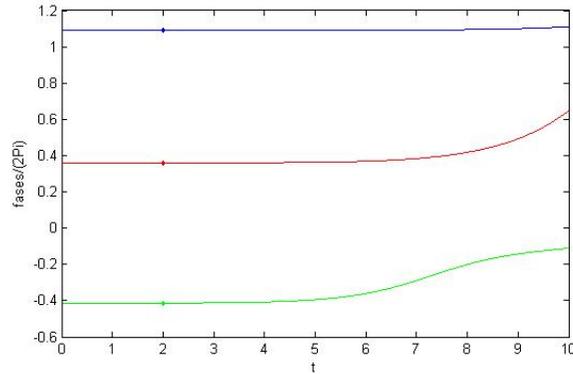


Figura 4.21: Componentes $\phi_i^{\Delta t}$ de la solución del modelo no lineal, cuando se aplican los controles de la figura 4.17.

Para controlar a través de todo el intervalo de tiempo hemos dividido éste en subintervalos de longitud $\Delta T = T/Q$, y denotamos $q\Delta T$ por T_q para $q = 1, \dots, Q$; entonces procedemos como sigue:

- Para $q = 0$, denotamos por \mathbf{y}_0 la diferencia $\phi_0 - \theta$, y resolvemos el problema lineal de control en $[0, T_1]$ ((4.4)-(4.6) para el caso cuando se controla con las tres juntas); denotamos por \mathbf{u}_1 el control correspondiente. Este control se impone como dato de entrada en el sistema 4.2, y denotamos por \mathbf{y}_1 la diferencia $\phi(T_1) - \theta$.
- Para $q > 0$, denotamos por \mathbf{y}_q la diferencia $\phi(T_q) - \theta$; resolvemos el problema lineal de control asociado en $[T_q, T_{q+1}]$ ((4.4)-(4.6) para el caso cuando se controla con las tres juntas), con \mathbf{y}_0 sustituido por \mathbf{y}_q , y denotamos por \mathbf{u}_{q+1} el control óptimo correspondiente. El control \mathbf{u}_{q+1} se impone en (4.2) y denotamos por \mathbf{y}_{q+1} la diferencia $\phi(T_{q+1}) - \theta$.

- Hacemos $q = q + 1$ y repetimos el proceso hasta que $T_Q = T$.

Hemos aplicado este método de particionar el tiempo al problema de control discreto asociado (4.51)-(4.53) que resulta cuando controlamos con las tres juntas, con $\phi_0 = \theta + \delta\theta$, donde θ y $\delta\theta$, al igual que los parámetros k_1 y k_2 , son los que se usaron para obtener las gráficas 4.17-4.19, pero con el intervalo total de tiempo $[0, 20]$; usamos $\Delta T = 2.0$. Después de $t = 20$, hemos tomado $\mathbf{u} = 0$ en (4.51)-(4.53) y en (4.2) para observar la evolución del sistema lineal tanto como del no lineal cuando se dejan de controlar. Los resultados se reportan en las figuras 4.22, 4.23 y 4.24. Observamos que los sistemas prácticamente se estabilizan para $1 \leq t \leq 20$, pero si se deja de controlar, los pequeños residuales de perturbación del sistema al tiempo $t = 20$, son suficientes para desestabilizarlo y lo inducen a transitar a un equilibrio estable en un lapso de tiempo finito.

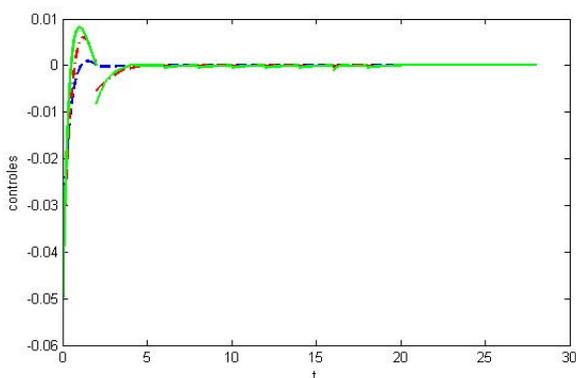


Figura 4.22: Controles calculados iterativamente cada 2 unidades de tiempo.

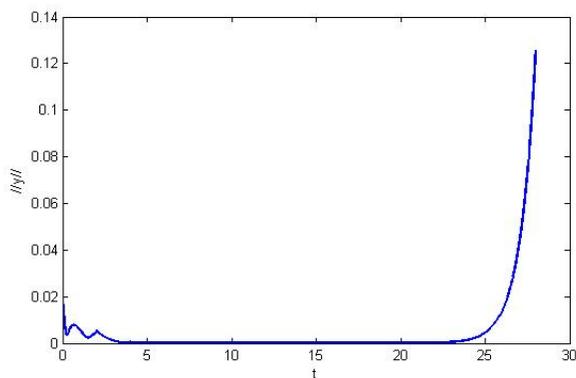


Figura 4.23: Norma euclidiana de la solución $\mathbf{y}^{\Delta t}$ usando los controles de la figura 4.22.

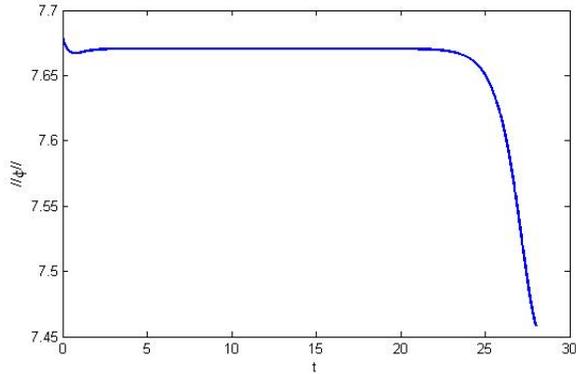


Figura 4.24: Norma euclidiana de la solución $\phi^{\Delta t}$ usando los controles de la figura 4.22.

4.6 Conclusiones

Se ha logrado estabilizar las fases del modelo AMJJ alrededor de un equilibrio inestable utilizando un enfoque clásico, esto es, linealizar el modelo alrededor del punto de equilibrio y controlarlo con el fin de estabilizarlo alrededor del equilibrio inestable. Una vez calculado el control para la versión lineal, se aplica el control al modelo no lineal, el cual en este caso, también resulta estabilizado alrededor del equilibrio inestable. Como el intervalo de tiempo en este tipo de fenómenos es grande, fue necesario dividir el intervalo original en subintervalos y calcular iterativamente el control lineal en cada subintervalo, obteniendo un control por pedazos que controla tanto el modelo lineal como el no lineal. Para el cálculo eficiente del control fue necesario formular el problema en forma operacional, donde el operador asociado es autoadjunto y elíptico, lo cual permite utilizar una versión del algoritmo de gradiente conjugado para funcionales cuadráticos. Los resultados numéricos muestran que para lograr la estabilización del modelo lineal, y por lo tanto del no lineal, es necesario controlar a través de al menos dos juntas de Josephson.

Capítulo 5

Conclusiones Generales

En esta tesis se han trabajado dos modelos matemáticos distintos, uno lineal en derivadas parciales usado para ajustar campos de viento, y el otro no lineal de ecuaciones diferenciales ordinarias. El primer problema consiste en generar, en un dominio dado, un campo de velocidad \mathbf{u} para un fluido incompresible, el cual ajusta a uno inicial \mathbf{u}^I obtenido por medidas experimentales, por ejemplo. El campo vectorial ajustado \mathbf{u} debe satisfacer conservación de masa y una condición de *no-flujo* en parte de la frontera. En el segundo problema se modela un circuito de tres juntas de Josephson acopladas inductivamente que se utiliza para controlar la transición del sistema entre dos estados de equilibrio, ya sea estables o inestables y para estabilizar el sistema físico alrededor de un punto de equilibrio inestable.

Aunque los modelos y los objetivos para los que se usan son muy distintos, ambos problemas se pueden plantear en un marco matemático común como problemas de control en ecuaciones diferenciales en donde las soluciones se pueden calcular por medio de cálculo variacional, técnicas de optimización y métodos de aproximación numérica. En particular, las soluciones en ambos problemas satisfacen criterios de optimalidad para funcionales cuadráticos. El problema de ajuste de campos de viento es un problema bien planteado al cual se le asocia una formulación de multiplicadores de Lagrange, mientras que el problema de las juntas de Josephson es un problema mal planteado al cual se le asocia una formulación de control usando penalización y/o regularización.

El enfoque usado para formular los modelos computacionales permite usar algoritmos y técnicas eficientes de control, optimización y cálculo numérico como gradiente conjugado, elemento finito, diferencias finitas y el método de Newton. Tanto en el caso de ajuste de campos de viento como en el de estabilización del modelo AMJJ alrededor de un equilibrio inestable, se estableció una ecuación funcional satisfecha por el multiplicador y por el control, respectivamente, lo que permite calcularlos eficientemente; en el caso de ajuste de campos de viento la contribución original de este trabajo consistió en la construcción de un preconditionador que resultó ser óptimo (natural) para resolver el problema de punto silla asociado en forma mucho más eficiente y precisa que los métodos tradicionales. Por otro lado, hasta donde sabemos, la aplicación de técnicas de control para hacer transitar el sistema de las juntas de Josephson entre dos equilibrios, y en particular para definir operadores de *Lectura/Escritura* de memoria en computación cuántica, también es una contribución original de este trabajo. Lo que normalmente han hecho otros autores es utilizar la técnica de intento-error, es decir, probar con diferentes pulsos gaussianos hasta encontrar los que producen los resultados deseados, lo cual obviamente no es eficiente.

Como posible trabajo futuro se contempla la aplicación de las juntas de Josephson a la computación cuántica, es decir, a la definición de una familia de qbits en términos de las fases de Josephson. Para el caso de campos de viento se considera trabajar en modelos tridimensionales además de introducir viscosidad (difusión) en el modelo. Estos modelos matemáticos y técnicas de solución también se pueden aplicar en otros problemas, como por ejemplo reconstrucción de imágenes cuando se tiene disponible sólo información parcial.

Apéndice A

Algoritmo de Gradiente Conjugado

A.1 Algoritmo de gradiente conjugado para funcionales cuadráticos

Consideremos la clase de problemas variacionales lineales siguiente:

$$\begin{cases} u \in \mathcal{V}, \\ a(u, v) = L(v), \forall v \in \mathcal{V}, \end{cases} \quad (\text{A.1})$$

donde

- (i) \mathcal{V} es un espacio de Hilbert real para el producto escalar (\cdot, \cdot) y la norma correspondiente $\|\cdot\|$;
- (ii) $a : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ es bilineal, continua, simétrica y \mathcal{V} -elíptica, i.e., $\exists \alpha > 0$ tal que $a(v, v) \geq \alpha \|v\|^2, \forall v \in \mathcal{V}$;
- (iii) $L : \mathcal{V} \rightarrow \mathbb{R}$ es lineal y continua.

Si las propiedades mencionadas se dan, entonces el problema (A.1) tiene una única solución (ver [34]); de hecho, la simetría de $a(\cdot, \cdot)$ no es necesaria para tener una solución única del problema (A.1); sin embargo, la simetría de $a(\cdot, \cdot)$, combinada con las otras propiedades, permite resolver el problema (A.1) con el algoritmo de gradiente conjugado siguiente:

- **Paso 1.** Se da $u^0 \in \mathcal{V}$.

- **Paso 2.** Resolver

$$\begin{cases} g^0 \in \mathcal{V}, \\ (g^0, v) = a(u^0, v) - L(v), \forall v \in \mathcal{V}, \end{cases} \quad (\text{A.2})$$

y hacer

$$w^0 = g^0. \quad (\text{A.3})$$

- **Paso 3.** Para $q \geq 0$, u^q , g^q y w^q conocidos, calcular u^{q+1} , g^{q+1} y w^{q+1} como sigue. Calcular

$$\rho^q = \|g^q\|^2 / a(w^q, w^q) \quad (\text{A.4})$$

y tomar

$$u^{q+1} = u^q - \rho^q w^q. \quad (\text{A.5})$$

Resolver

$$\begin{cases} g^{q+1} \in \mathcal{V}, \\ (g^{q+1}, v) = (g^q, v) - \rho^q a(w^q, v), \forall v \in \mathcal{V}, \end{cases} \quad (\text{A.6})$$

y calcular

$$\gamma^q = \|g^{q+1}\|^2 / \|g^q\|^2, \quad (\text{A.7})$$

$$w^{q+1} = g^{q+1} + \gamma^q w^q \quad (\text{A.8})$$

- **Paso 4.** Hacer $n=n+1$ e ir a (A.4).

A.2 Algoritmo de gradiente conjugado para funcionales no cuadráticos en espacios de Hilbert

Consideremos el siguiente problema de minimización sin restricciones

$$\begin{cases} u \in V, \\ J(u) \leq J(v), \forall v \in V. \end{cases} \quad (\text{A.9})$$

donde

- V es un espacio de Hilbert real para el producto interior (\cdot, \cdot) y la norma asociada $\|\cdot\|$.
- $J : V \rightarrow \mathbb{R}$ es diferenciable sobre V ; denotamos por DJ su diferencial.

Supóngase que el problema (A.9) tiene una solución; esta solución verifica

$$DJ(u) = 0, \quad (\text{A.10})$$

o equivalentemente

$$\begin{cases} u \in V, \\ \langle DJ(u), v \rangle = 0, \forall v \in V, \end{cases} \quad (\text{A.11})$$

donde en (A.11) los símbolos $\langle \cdot, \cdot \rangle$ denotan el par de dualidad entre V' (el espacio dual de V) y V . De acuerdo al capítulo 3 de [23], un algoritmo de gradiente conjugado para la solución del problema (A.9) es como sigue:

- Suponer que

$$u^0 \text{ es dado en } V \text{ (} u^0 = 0 \text{ por ejemplo)}. \quad (\text{A.12})$$

- Resolver

$$\begin{cases} g^0 \in V, \\ (g^0, v) = \langle DJ(u^0), v \rangle, \forall v \in V. \end{cases} \quad (\text{A.13})$$

- Si $\frac{\|g^0\|}{\max\{1, \|u^0\|\}} \leq tol$, tomar $u = u^0$; en otro caso, hacer

$$w^0 = g^0. \quad (\text{A.14})$$

Para $q \geq 0$, u^q , g^q y w^q conocidos, los últimos dos diferentes de 0, calculamos u^{q+1} , y de ser necesario w^{q+1} como sigue:

- Resolver

$$\begin{cases} \rho_q \in \mathbb{R}, \\ J(u^q - \rho_q w^q) \leq J(u^q - \rho w^q), \forall \rho \in \mathbb{R}. \end{cases} \quad (\text{A.15})$$

- Tomar

$$u^{q+1} = u^q - \rho_q w^q. \quad (\text{A.16})$$

- Resolver

$$\begin{cases} g^{q+1} \in V, \\ (g^{q+1}, v) = \langle DJ(u^{q+1}), v \rangle, \forall v \in V. \end{cases} \quad (\text{A.17})$$

- Si $\frac{\|g^{q+1}\|}{\max\{\|g^0\|, \|u^{q+1}\|\}} \leq tol$, tomar $u = u^{q+1}$; en otro caso, calcular

$$\beta_q = \frac{\|g^{q+1}\|^2}{\|g^q\|^2} \quad (\text{A.18})$$

y

$$w^{q+1} = g^{q+1} + \beta_q w^q. \quad (\text{A.19})$$

- Hacer $q + 1 \rightarrow q$ y regresar a (A.15) .
- Fin del algoritmo.

Nota 6.1. En muchas aplicaciones, la parte mas delicada del algoritmo (A.12)-(A.19) no es la solución de los problemas variacionales lineales (A.13) y (A.17) sino, de hecho, la solución del problema unidimensional (A.15) (conocido en optimización como búsqueda en línea). Suponiendo que J tiene regularidad C^2 sobre V vamos a discutir enseguida la solución de (A.15) por un método de Newton:

Supongamos que ρ_* es solución del problema de minimización (A.15). Se sigue de la regla de la cadena (después de eliminar los superíndices q , por simplicidad) que

$$\langle DJ(u - \rho_* w), w \rangle = 0. \quad (\text{A.20})$$

Definamos $H : \mathbb{R} \rightarrow \mathbb{R}$ por

$$H(\rho) = \langle DJ(u - \rho w), w \rangle; \quad (\text{A.21})$$

tenemos entonces

$$H(\rho_*) = 0, \quad (\text{A.22})$$

y

$$H'(\rho) = - \langle D^2 J(u - \rho w) w, w \rangle. \quad (\text{A.23})$$

Aplicando el método de Newton a la solución de (A.22) da

- Suponer que

$$\rho^0 \text{ es dado en } \mathbb{R} \text{ (} \rho^0 = 0 \text{ parece ser un buen candidato en este contexto)}. \quad (\text{A.24})$$

- Para $m \geq 0$, ρ^{m+1} se calcula por medio de

$$\rho^{m+1} = \rho^m - \frac{H(\rho^m)}{H'(\rho^m)}. \quad (\text{A.25})$$

Una formulación mas explícita para (A.25) es

$$\rho^{m+1} = \rho^m + \frac{\langle DJ(u - \rho^m w), w \rangle}{\langle D^2 J(u - \rho^m w) w, w \rangle}. \quad (\text{A.26})$$

Antes de usar el método de Newton uno puede tratar las siguientes opciones, ordenadas de acuerdo a su dificultad:

- Usar como caja negra alguna rutina que no use derivadas para la minimización de funciones en una variable. Tales rutinas están disponibles en Matlab (función `fminbnd.m`, por ejemplo). Algunos métodos que no usan derivadas se presentan en [35].
- Usar un método de extrapolación cúbica con backtracking, como se discute en [36].
- Usar un método de la secante.

En nuestra implementación nosotros usamos el método de Newton y la función `fminbnd.m` de MatLab.

Apéndice B

Método de Newton para resolver los problemas de búsqueda en línea

Consideramos aquí los detalles para aplicar el método de Newton para resolver los problemas de búsqueda en línea que aparecen dentro del algoritmo de gradiente conjugado para resolver los problemas de minimización considerados en esta tesis. Se describe tanto el caso continuo como su versión discreta.

B.1 El problema de minimización continuo

El problema de minimización en que estamos interesados es

$$\begin{cases} u \in \mathcal{U}_{ad} = L^2(0, T), \\ J(u) \leq J(v), \forall v \in \mathcal{U}_{ad}, \end{cases} \quad (\text{B.1})$$

donde (con $k > 0$ y $\|\cdot\|$ la norma euclídeana canónica)

$$J(v) = \frac{1}{2} \int_0^T v^2 dt + \frac{k}{2} \|\mathbf{y}(T) - \mathbf{y}_T\|^2, \quad (\text{B.2})$$

y la función vectorial $\mathbf{y} = \{y_1, y_2, y_3\}$ es la solución del siguiente problema de valor inicial:

$$\begin{cases} \gamma_1 \frac{dy_1}{dt} + \kappa_1(y_1 - y_2) + \sin y_1 = i_1 + v \text{ en } (0, T), \\ \gamma_2 \frac{dy_2}{dt} + \kappa_1(y_2 - y_1) + \kappa_2(y_2 - y_3) + \sin y_2 = i_2 \text{ en } (0, T), \\ \gamma_3 \frac{dy_3}{dt} + \kappa_2(y_3 - y_2) + \sin y_3 = i_3 \text{ en } (0, T). \\ \mathbf{y}(0) = \mathbf{y}_0. \end{cases} \quad (\text{B.3})$$

Si $\mathbf{p} = \{p_1, p_2, p_3\}$ se define como la solución del siguiente problema (**sistema adjunto**):

$$\begin{cases} -\Gamma \frac{d\mathbf{p}}{dt} + K\mathbf{p} + \begin{pmatrix} \cos y_1 & 0 & 0 \\ 0 & \cos y_2 & 0 \\ 0 & 0 & \cos y_3 \end{pmatrix} \mathbf{p} = \mathbf{0}, \text{ en } (0, T) \\ \Gamma \mathbf{p}(T) = k(\mathbf{y}(T) - \mathbf{y}_T). \end{cases} \quad (\text{B.4})$$

entonces

$$DJ(v) = v + p_1. \quad (\text{B.5})$$

B.2 El problema de búsqueda en línea asociado

Dentro del algoritmo de gradiente conjugado para resolver (B.1) debemos resolver en cada paso un problema de búsqueda en línea del tipo

$$\begin{cases} \rho_* \in \mathbb{R}, \\ J(v - \rho_* w) \leq J(v - \rho w), \forall \rho \in \mathbb{R}. \end{cases} \quad (\text{B.6})$$

con v y w fijos en $L^2(0, T)$.

B.3 Resolviendo (B.6) con el método de Newton

Supongamos que ρ_* es solución de (B.6). Se sigue de la regla de la cadena que

$$\langle DJ(v - \rho_* w), w \rangle = 0. \quad (\text{B.7})$$

Definimos $H : \mathbb{R} \rightarrow \mathbb{R}$ por

$$H(\rho) = \langle DJ(v - \rho w), w \rangle; \quad (\text{B.8})$$

tenemos entonces

$$H(\rho_*) = 0, \quad (\text{B.9})$$

y

$$\dot{H}(\rho) = - \langle D^2 J(v - \rho w) w, w \rangle. \quad (\text{B.10})$$

Dado ρ^0 ($\rho^0 = 0$ parece ser una buena opción en este contexto), para $m \geq 0$ la solución por el método de Newton para (B.9) queda

$$\rho^{m+1} = \rho^m - \frac{H(\rho^m)}{\dot{H}(\rho^m)}. \quad (\text{B.11})$$

o mas explícitamente

$$\rho^{m+1} = \rho^m + \frac{\langle DJ(v - \rho w), w \rangle}{\langle D^2 J(v - \rho w) w, w \rangle}. \quad (\text{B.12})$$

Como estamos trabajando con $\mathcal{U}_{ad} = L^2(0, T)$, donde los pares de dualidad coinciden con los productos interiores, podemos escribir

$$H(\rho) = \int_0^T DJ(v - \rho w) w \, dt; \quad (\text{B.13})$$

con

$$DJ(v - \rho w) = v - \rho w + p_{1\rho}. \quad (\text{B.14})$$

donde $p_{1\rho}$ se obtiene resolviendo el problema

$$\left\{ \begin{array}{l} \gamma_1 \frac{dy_{1\rho}}{dt} + \kappa_1(y_{1\rho} - y_{2\rho}) + \sin y_{1\rho} = i_1 + v - \rho w \text{ en } (0, T), \\ \gamma_2 \frac{dy_{2\rho}}{dt} + \kappa_1(y_{2\rho} - y_{1\rho}) + \kappa_2(y_{2\rho} - y_{3\rho}) + \sin y_{2\rho} = i_2 \text{ en } (0, T), \\ \gamma_3 \frac{dy_{3\rho}}{dt} + \kappa_2(y_{3\rho} - y_{2\rho}) + \sin y_{3\rho} = i_3 \text{ en } (0, T). \\ \mathbf{y}_\rho(0) = \mathbf{y}_0. \end{array} \right. \quad (\text{B.15})$$

y luego el problema

$$\left\{ \begin{array}{l} -\Gamma \frac{d\mathbf{p}_\rho}{dt} + K\mathbf{p}_\rho + \begin{pmatrix} \cos y_{1\rho} & 0 & 0 \\ 0 & \cos y_{2\rho} & 0 \\ 0 & 0 & \cos y_{3\rho} \end{pmatrix} \mathbf{p}_\rho = \mathbf{0}, \text{ en } (0, T) \\ \Gamma \mathbf{p}_\rho(T) = k(\mathbf{y}_\rho(T) - \mathbf{y}_T). \end{array} \right. \quad (\text{B.16})$$

Solo falta calcular $\dot{H}(\rho) = \frac{dH}{d\rho}$. Esto es casi directo, ya que por la definición de H (y por la regla de la cadena) tenemos

$$\dot{H}(\rho) = \int_0^T [-w + \dot{p}_{1\rho}] w \, dt,$$

donde $\dot{p}_{1\rho}$ se obtiene resolviendo el problema

$$\left\{ \begin{array}{l} -\Gamma \frac{d\dot{\mathbf{y}}_\rho}{dt} + K\dot{\mathbf{y}}_\rho + \begin{pmatrix} \cos y_{1\rho} & 0 & 0 \\ 0 & \cos y_{2\rho} & 0 \\ 0 & 0 & \cos y_{3\rho} \end{pmatrix} \dot{\mathbf{y}}_\rho = \begin{pmatrix} -w \\ 0 \\ 0 \end{pmatrix}, \text{ en } (0, T) \\ \mathbf{y}_\rho(0) = \mathbf{y}_0. \end{array} \right. \quad (\text{B.17})$$

y luego el problema

$$\left\{ \begin{array}{l} -\Gamma \frac{d\dot{\mathbf{p}}_\rho}{dt} + K\dot{\mathbf{p}}_\rho + \begin{pmatrix} \cos y_{1\rho} & 0 & 0 \\ 0 & \cos y_{2\rho} & 0 \\ 0 & 0 & \cos y_{3\rho} \end{pmatrix} \dot{\mathbf{p}}_\rho = \\ \begin{pmatrix} \dot{y}_{1\rho} \sin y_{1\rho} & 0 & 0 \\ 0 & \dot{y}_{2\rho} \sin y_{2\rho} & 0 \\ 0 & 0 & \dot{y}_{3\rho} \sin y_{3\rho} \end{pmatrix} \mathbf{p}_\rho, \text{ en } (0, T) \\ \Gamma \dot{\mathbf{p}}_\rho(T) = k\dot{\mathbf{y}}_\rho(T). \end{array} \right. \quad (\text{B.18})$$

B.4 Discretización del problema (B.1)

Aproximamos (B.1) por

$$\left\{ \begin{array}{l} \mathbf{u}^{\Delta t} \in \mathcal{U}_{ad}^{\Delta t}, \\ J^{\Delta t}(\mathbf{u}^{\Delta t}) \leq J^{\Delta t}(\mathbf{v}), \forall \mathbf{v} \in \mathcal{U}_{ad}^{\Delta t}, \end{array} \right. \quad (\text{B.19})$$

donde:

- $\Delta t = T/N$ con N un entero positivo "grande".
- $\mathcal{U}_{ad}^{\Delta t} = \mathbb{R}^N$.

El funcional de costo $J^{\Delta t}$ es definido por

$$J^{\Delta t}(\mathbf{v}) = \frac{\Delta t}{2} \sum_{n=1}^N |v^n|^2 + \frac{k}{2} \|\mathbf{y}^N - \mathbf{y}_T\|^2,$$

con $\mathbf{v} = \{v^n\}_{n=1}^N$ y $\{\mathbf{y}^n\}_{n=1}^N$ obtenido de \mathbf{v} y \mathbf{y}_0 a través de la siguiente variante discreta de (B.3):

$$\mathbf{y}^0 = \mathbf{y}_0, \quad (\text{B.20})$$

y para $n = 1, \dots, N$

$$\left\{ \Gamma \frac{\mathbf{y}^n - \mathbf{y}^{n-1}}{\Delta t} + K\mathbf{y}^n + \begin{pmatrix} \sin y_1^{n-1} \\ \sin y_2^{n-1} \\ \sin y_3^{n-1} \end{pmatrix} = \begin{pmatrix} f_1 + v^n \\ f_2 \\ f_3 \end{pmatrix} \text{ en } (0, T), \right.$$

Si en $\mathcal{U}_{ad}^{\Delta t} = \mathbb{R}^N$ consideramos el producto interior

$$(\mathbf{v}, \mathbf{w})_{\Delta t} = \Delta t \sum_{n=1}^N v^n w^n, \quad \forall \mathbf{v} = \{v^n\}_{n=1}^N, \mathbf{w} = \{w^n\}_{n=1}^N \in \mathbb{R}^N,$$

y definimos $\{\mathbf{p}^n\}_{n=1}^{N+1}$ como la solución del siguiente problema adjunto:

$$\Gamma \mathbf{p}^{N+1} = k(\mathbf{y}^N - \mathbf{y}_T), \quad (\text{B.21})$$

$$\Gamma \frac{\mathbf{p}^N - \mathbf{p}^{N+1}}{\Delta t} + K \mathbf{p}^N = \mathbf{0}, \quad (\text{B.22})$$

y para $n = N - 1, \dots, 1$:

$$\Gamma \frac{\mathbf{p}^n - \mathbf{p}^{n+1}}{\Delta t} + K \mathbf{p}^n + \begin{pmatrix} \cos y_1^n & 0 & 0 \\ 0 & \cos y_2^n & 0 \\ 0 & 0 & \cos y_3^n \end{pmatrix} \mathbf{p}^{n+1} = \mathbf{0}, \quad (\text{B.23})$$

entonces

$$DJ^{\Delta t}(\mathbf{v}) = \{v^n + p_1^n\}_{n=1}^N. \quad (\text{B.24})$$

B.5 El problema de búsqueda en línea asociado a (B.19)

Como en el texto de la tesis, usaremos la notación $\mathbf{y}_q^n = \{y_{iq}^n\}_{i=1}^3$ para denotar el valor discreto de la función vectorial \mathbf{y} al tiempo $n\Delta t$ y en la iteración q ; similarmente, u_q^n denota el valor discreto del control u al tiempo $n\Delta t$ e iteración q . Con esto, los problemas de búsqueda en línea asociados con el algoritmo de gradiente conjugado para resolver el problema finito-dimensional (B.19) son del tipo

$$\begin{cases} \rho_* \in \mathbb{R}, \\ J^{\Delta t}(\mathbf{v} - \rho_* \mathbf{w}) \leq J^{\Delta t}(\mathbf{v} - \rho \mathbf{w}), \forall \rho \in \mathbb{R}. \end{cases} \quad (\text{B.25})$$

con \mathbf{v} y \mathbf{w} fijos en \mathbb{R}^N .

B.6 Resolviendo (B.25) con el método de Newton

Si $\rho_* \in \mathbb{R}$ es solución de (B.25), entonces ρ_* es un cero de la función

$$H(\rho) = \Delta t \sum_{n=1}^N [(v^n - \rho w^n) + p_{1\rho}^n] w^n. \quad (\text{B.26})$$

donde $\{p_{1\rho}^n\}_{n=1}^{N+1}$ se define calculando la solución $\{\mathbf{y}_\rho^n\}_{n=1}^N = \{(y_{1\rho}^n, y_{2\rho}^n, y_{3\rho}^n)\}_{n=1}^N$ de

$$\begin{cases} \mathbf{y}_\rho^0 = \mathbf{y}_0, \\ \text{para } n = 1, \dots, N \text{ resolver} \\ \Gamma \frac{\mathbf{y}_\rho^n - \mathbf{y}_\rho^{n-1}}{\Delta t} + K \mathbf{y}_\rho^n + \begin{pmatrix} \sin y_{1\rho}^{n-1} \\ \sin y_{2\rho}^{n-1} \\ \sin y_{3\rho}^{n-1} \end{pmatrix} = \begin{pmatrix} i_1 + v^n - \rho w^n \\ 0 \\ 0 \end{pmatrix}, \end{cases} \quad (\text{B.27})$$

y entonces resolver para $\{\mathbf{p}_\rho^n\}_{n=1}^{N+1} = \{(p_{1\rho}^n, p_{2\rho}^n, p_{3\rho}^n)\}_{n=1}^{N+1}$ el sistema

$$\left\{ \begin{array}{l} \Gamma \mathbf{p}_\rho^{N+1} = k(\mathbf{y}_\rho^N - \mathbf{y}_T), \\ \Gamma \frac{\mathbf{p}_\rho^N - \mathbf{p}_\rho^{N+1}}{\Delta t} + K \mathbf{p}_\rho^N = \mathbf{0}, \\ \text{para } n = N-1, \dots, 1 \text{ resolver} \\ \Gamma \frac{\mathbf{p}_\rho^n - \mathbf{p}_\rho^{n+1}}{\Delta t} + K \mathbf{p}_\rho^n \\ + \begin{pmatrix} \cos y_{1\rho}^n & 0 & 0 \\ 0 & \cos y_{2\rho}^n & 0 \\ 0 & 0 & \cos y_{3\rho}^n \end{pmatrix} \mathbf{p}_\rho^{n+1} = \mathbf{0}. \end{array} \right. \quad (\text{B.28})$$

Para resolver la ecuación $H(\rho_*) = 0$ con Newton se requiere el conocimiento de $\dot{H}(\rho)$: observando (B.26) concluimos que

$$\dot{H}(\rho) = \Delta t \sum_{n=1}^N [-w^n + \dot{p}_{1\rho}^n] w^n, \quad (\text{B.29})$$

con $\dot{p}_{1\rho}^n$ obtenido al diferenciar (B.27) y (B.28) con respecto a ρ , esto es, al resolver para $\dot{\mathbf{y}}_\rho^n$

$$\left\{ \begin{array}{l} \dot{\mathbf{y}}_\rho^0 = \mathbf{0}, \\ \text{para } n = 1, \dots, N \text{ resolver} \\ \Gamma \frac{\dot{\mathbf{y}}_\rho^n - \dot{\mathbf{y}}_\rho^{n-1}}{\Delta t} + K \dot{\mathbf{y}}_\rho^n + \begin{pmatrix} \cos y_{1\rho}^{n-1} \\ \cos y_{2\rho}^{n-1} \\ \cos y_{3\rho}^{n-1} \end{pmatrix} \dot{\mathbf{y}}_\rho^{n-1} = \begin{pmatrix} -w^n \\ 0 \\ 0 \end{pmatrix}, \end{array} \right. \quad (\text{B.30})$$

y luego resolver para \mathbf{p}_ρ^n :

$$\left\{ \begin{array}{l} \Gamma \mathbf{p}_\rho^{N+1} = k \dot{\mathbf{y}}_\rho^N, \\ \Gamma \frac{\mathbf{p}_\rho^N - \mathbf{p}_\rho^{N+1}}{\Delta t} + K \mathbf{p}_\rho^N = \mathbf{0}, \\ \text{para } n = N-1, \dots, 1 \text{ resolver} \\ \Gamma \frac{\mathbf{p}_\rho^n - \mathbf{p}_\rho^{n+1}}{\Delta t} + K \mathbf{p}_\rho^n \\ + \begin{pmatrix} \cos y_{1\rho}^n & 0 & 0 \\ 0 & \cos y_{2\rho}^n & 0 \\ 0 & 0 & \cos y_{3\rho}^n \end{pmatrix} \mathbf{p}_\rho^{n+1} = \\ \begin{pmatrix} \dot{y}_{1\rho}^n \sin y_{1\rho}^n & 0 & 0 \\ 0 & \dot{y}_{2\rho}^n \sin y_{2\rho}^n & 0 \\ 0 & 0 & \dot{y}_{3\rho}^n \sin y_{3\rho}^n \end{pmatrix} \mathbf{p}_\rho^{n+1}. \end{array} \right. \quad (\text{B.31})$$

Con esto podemos ya aplicar Newton para resolver (B.25): Dado ρ^0 ($\rho^0 = 0$ parece ser una buena opción en este contexto), iterar con

$$\rho^{m+1} = \rho^m - \frac{H(\rho^m)}{\dot{H}(\rho^m)}. \quad (\text{B.32})$$

Bibliografía

- [1] C. A. Sherman, A mass-consistent model for wind fields over complex terrain, *J. Appl. Meteor.* 17 (1978) 312–319.
- [2] S. Finardi, G. Tinarelli, A. Nanni, G. Brusasca, G. Carboni, Evaluation of a 3-d flow and pollutant dispersion modelling system to estimate climatological ground level concentrations in complex coastal sites, *International Journal of Environment and Pollution* 16 (2001) 472–482.
- [3] A. Oliver, G. Montero, R. Montenegro, E. Rodríguez, J. M. Escobar, A. Pérez-Foguet, Adaptive finite element simulation of stack pollutant emissions over complex terrains, *Energy* 49 (2013) 47–60.
- [4] F. Castino, L. Rusca, G. Solari, Wind climate micro-zoning: a pilot application to Liguria region (North-Western Italy), *Journal of Wind Engineering and Industrial Aerodynamics* 91 (2003) 1353–1375.
- [5] C. F. Ratto, An overview of mass-consistent models, in: D. P. Lalas, C. F. Ratto (Eds.), *Modelling of Atmosphere Flow Fields*, World Scientific Publications, 1996, pp. 379–400.
- [6] Y. Sasaki, An objective analysis based on the variational method, *Journal Met. Soc. Japan* 36 (1958) 77–88.
- [7] W. T. Pennel, An evaluation of the role of numerical wind field models in wind turbine siting, Tech. Rep. PNL-SA-11129, Battelle Memorial Institute, Pacific Northwest Laboratory, Richland, Washington (1983).
- [8] Y. Wang, C. Williamson, D. Garvey, S. Chang, J. Cogan, Application of a multigrid method to a mass-consistent diagnostic wind model, *Journal of Applied Meteorology* 44 (2005) 1078–1089.
- [9] D. G. Ross, I. N. Smith, P. C. Manins, D. G. Fox, Diagnostic wind field modelling for complex terrain: Model development and testing, *J. Appl. Meteor.* 27 (1988) 785–796.
- [10] M. A. Nunez, New scheme to compute mass-consistent models of geophysical flows, in: Petrópolis-R.J. (Ed.), *Proceedings of the I Workshop on Asymptotics for Parabolic and Hyperbolic Systems*, Laboratório Nacional de Computação Científica, Brasil, 2008.
- [11] L. Ferragut, R. Montenegro, G. Montero, E. Rodríguez, M. Asensio, J. Escobar, Comparison between 2.5-D and 3-D realistic models for wind field adjustment, *J. Wind. Eng. Ind. Aerodyn.* 98 (10–11) (2010) 548–558.

- [12] D. A. Cervantes, P. González-Casanova, C. Gout, L. Juárez, L. Reséndiz, Vector field approximation using radial basis functions, *Journal of Computational and Applied Mathematics* 240 (2013) 163–173.
- [13] A. H. Alaoui, L. White, A. Shapiro, Estimation of vertical wind velocity using doppler radar data with vorticity constraints, *Applied Mathematics and Computation* 215 (2010) 4119–4131.
- [14] F. B. Hacene, M. T. Abbes, N. K. Merzouk, L. Loukarfi, H. Mahmoudi, M. F. A. Goosen, Development of a simulation model for a three-dimensional wind velocity field using tenes algeria as a case study, *Renewable and Sustainable Energy Reviews* 16 (2012) 29–36.
- [15] G. G. Koch, A. Benzing, C. P. Mayer, An approach for urban sensing with quality-aware situation detection and efficient communication, *Electronic Communications of the EASST* 37 (2011) 1–10.
- [16] J. Cascón, L. Ferragut, An adaptive mixed finite element method for wind field adjustment, *Advances in Numerical Methods for Environmental Engineering* 38 (2007) 350–357.
- [17] D. W. Pepper, X. Wang, An h-adaptive finite-element technique for constructing 3D wind fields, *Journal of Applied Meteorology and Climatology* 48 (2009) 580–599.
- [18] L. Yu-Chieng, C. Shao-Fan, S. Juanzhen, An application of the immersed boundary method for recovering the three-dimensional wind fields over complex terrain using multiple-doppler radar data, *Monthly Weather Review* 140 (5) (2012) 1603–1619.
- [19] L. Ferragut, M. I. Asensio, J. Simon, High definition local adjustment model of 3D wind fields performing only 2D computations, *Communications in Numerical Methods in Engineering* 27 (2011) 510–523.
- [20] L. Bonaventura, A. Iske and E. Miglio, Kernel-Based Vector Field Reconstruction in Computational Fluid Dynamic Models, *Int. J. Numer. Meth. Fluids*, 66 (6) (2011) 714–729.
- [21] P. S. Peixoto, S. R. M. Barros, On vector field reconstructions for semi-Lagrangian transport methods on geodesic staggered grids, *J. of Comput. Physics*, 273 (2014) 185–211.
- [22] J. L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York, NY, 1971.
- [23] R. Glowinski, Finite element methods for incompressible viscous flow. In *Handbook of Numerical Analysis*, Vol. IX, P.G. Ciarlet & J.L. Lions, eds., North-Holland, Amsterdam, 2003, 3-1176.
- [24] C. F. Flores, L. H. Juárez, M. A. Nunez, M. L. Sandoval, Algorithms for vector field generation in mass consistent models, *Journal of Numerical Methods for Partial Differential Equations* 26 (4) (2009) 826–842.
- [25] M. Bercovier, O. Pironneau, Error estimates for the finite element method solution of the stokes problem in the primitive variables, *Numer. Math.* 33 (1979) 211–224.

- [26] T. Kitada, A. Kaki, H. Ueda, L. K. Peters, Estimation of the vertical air motion from limited horizontal wind data—a numerical experiment, *Atmos. Environ.* 17 (1983) 2181–2192.
- [27] G. F. Homicz, Three-dimensional wind field modelling: A review, Tech. Rep. report SAN2002–2597, Sandia National Laboratories (2002).
- [28] S. T. Wereley, C. D. Meinhart, Recent advances in micro-particle image velocimetry, *Annu. Rev. Fluid Mech.* 42 (2010) 557–576.
- [29] D. Sun, S. Roth, M. J. Black, A quantitative analysis of current practices in optical flow estimation and the principles behind them, *International Journal of Computer Vision* 106 (2014) 510–523.
- [30] F. Bedard, N. Welker, G. R. Cotter, M. A. Escavage and J. T. Pinkston, Superconducting Technology Assessment, National Security Agency, Office of Corporate Assessment (2005).
- [31] F. A. Holmes, L. Ripple, M. A. Manheimer, Energy-Efficient Superconducting Computing-Power Budgets and Requirements, *IEEE Transactions on Applied Superconductivity* 23(3) (2013).
- [32] IARPA, Broad Agency Announcement: IARPA-BAA-13-05, Cryogenic Computing Complexity (C3) Program (2013).
- [33] Y. Braiman and B. Neschke, Josephson Junction Array Cryogenic Memory. Internal Report, U. of Tennessee (2014).
- [34] R. Glowinski, J.L. Lions and J. W. He, Exact and Approximate Controllability for Distributed Parameter Systems: A Numerical Approach (*Encyclopedia of Mathematics and its Applications*). Cambridge University Press (2008).
- [35] R. P. Brent, Algorithms for minimization without derivatives. Courier Dover Publications (1973).
- [36] J. E. Dennis and R. B. Schnabel, Numerical Methods for Unconstrained Optimization and nonlinear equation, SIAM, Vol 16 (1996).
- [37] J. L. Lions, Optimal Control of Systems Governed by Partial Differential Equations. Springer Verlag, New York, New York (1971).