

**UNA ESTRATEGIA INSPIRADA EN CONTROL PARA VARIAR
EL PARÁMETRO DE REINICIO m DE GMRES(m)**

Rolando Cuevas Núñez

Orientadores: Prof. Christian E. Schaerer Serra, D.Sc.
Prof. Amit Bhaya, Ph.D.

Tesis presentada a la Facultad Politécnica, Universidad Nacional de Asunción,
como requisito para la obtención del Grado de Máster en Informática.

ASUNCIÓN - PARAGUAY

Diciembre - 2010

UNA ESTRATEGIA INSPIRADA EN CONTROL PARA VARIAR
EL PARÁMETRO DE REINICIO m DE GMRES(m)

Rolando Cuevas Núñez

Aprobado en Diciembre del 2010.

Prof. Benjamín Barán, D.Sc.

Prof. Amit Bhaya, Ph.D.

Prof. Horacio Legal, D.Sc.

Prof. Miki Saito, Dra.Ing.

Prof. Christian E. Schaerer, D.Sc.

Datos internacionales de Catalogación en la Publicación (CIP)
DE BIBLIOTECA CENTRAL DE LA UNA

Cuevas Núñez, Rolando

UNA ESTRATEGIA INSPIRADA EN CONTROL PARA VARIAR EL
PARÁMETRO DE REINICIO m DE GMRES(m)/Rolando Cuevas Núñez. –
Asunción, 2010.

112 p.: il.; 29,7cm.

Tesis (Maestría en Informática) – Facultad Politécnica, 2010.

REFERENCIAS BIBLIOGRÁFICAS: p. 90 – 93.

1. cambio de parámetro de reinicio. 2. GMRES con reinicio. 3. control.
I. Título.

A mis padres.

Agradecimientos

Al LCCA (Laboratorio de Computación Científica y Aplicada) de la Facultad Politécnica de la Universidad Nacional de Asunción.

A la Facultad Politécnica por facilitar mi viaje a la Universidad Federal de Rio de Janeiro (UFRJ) para trabajar con el Prof. Bhaya.

Al NACAD (Núcleo de Computación de Alto Desempeño) de la UFRJ, por haberme recibido durante mi estancia en Rio de Janeiro.

A mis orientadores: los profesores Ds. Sc. Schaerer y Ph. D. Bhaya, con su guía este trabajo ha finalizado exitosamente.

**UNA ESTRATEGIA INSPIRADA EN CONTROL PARA VARIAR
EL PARÁMETRO DE REINICIO m DE GMRES(m)**

Autor: Rolando Cuevas Núñez
Orientadores: Christian E. Schaerer Serra
Amit Bhaya

RESUMEN

El método del Residuo Mínimo Generalizado (GMRES) es uno de los algoritmos más populares de entre los que utilizan subespacios de Krylov para la resolución de sistemas lineales de ecuaciones $Ax = b$ cuyas matrices son grandes, dispersas (poco densas) y no-simétricas o no-hermitianas. Debido al costo computacional de la construcción de una base ortogonal para un subespacio de Krylov, en la práctica se utiliza el método GMRES con reinicio (denotado por GMRES(m)). Éste construye en cada ciclo una base de tamaño m para un subespacio de Krylov, donde se prefiere un valor de m pequeño en comparación con la dimensión de la matriz. Sin embargo, si el m apropiado no es elegido, la convergencia del algoritmo GMRES(m) no está garantizada y el método puede experimentar convergencia lenta o estancamiento. Desafortunadamente, es difícil saber a priori qué valor de m utilizar y varios investigadores han propuesto diferentes formas adaptativas para elegirlo. En esta tesis consideramos GMRES(m) como un problema de control en el que el parámetro de reinicio m es elegido de forma adaptativa en cada ciclo j utilizando una regla inspirada en un controlador proporcional derivativo de la forma:

$$m_{j+1} = m_j - \alpha \frac{\|r_j\|}{\|r_{j-1}\|} + \beta \frac{\|r_j\| - \|r_{j-2}\|}{2\|r_{j-1}\|},$$

donde α y β son elegidos de forma experimental. La ventaja de este método llamado PD-GMRES, es que solo se necesitan algunos vectores adicionales y el controlador tiene la capacidad de cambiar (aumentar o disminuir) la dimensión del subespacio de Krylov si detecta algún problema de convergencia.

Realizamos comparaciones numéricas entre PD-GMRES y las propuestas de Baker [BJK09] y Gonçalez [GC05] para la elección del valor de m . Los problemas de prueba, elegidos del Repositorio de Matrices de la Universidad de Florida [bTDfUoFfAR], fueron: (1) add20, (2) cdde1, (3) circuit, (4) fpga.trans_01, (5)

orsirr_1, (6) orsreg_1, (7) pde2961, (8) raefsky1, (9) raefsky2, (10) rdb2048, (11) sherman4, (12) steam2, (13) wang2, (14) watt_1, (15) young3c. Para problemas sin un lado derecho b , el mismo fue generado usando una distribución uniforme con valores entre el mínimo y el máximo de los valores de A . Los experimentos numéricos llevados a cabo en el conjunto de prueba muestran que PD-GMRES tiene una mejor tasa de convergencia que GMRES(30) excepto en dos problemas (problemas 7 y 10). Es importante remarcar que el algoritmo propuesto PD-GMRES es fácil de implementar y ha demostrado tener buena convergencia en este conjunto de prueba.

Palabras clave: *GMRES(m) adaptativo, controlador PD (proporcional derivativo), formulación de control*

**A CONTROL-INSPIRED STRATEGY FOR VARYING THE
RESTART PARAMETER m OF GMRES(m)**

Author: Rolando Cuevas Núñez
Advisors: Christian E. Schaerer Serra
Amit Bhaya

SUMMARY

The Generalized Minimal Residual (GMRES) method is one of the most popular algorithms, of the class of Krylov subspace methods, for the solution of large, sparse and unsymmetric linear systems of equations $Ax = b$. Due to the computational cost of the construction of an orthogonal basis for a Krylov subspace, the restarted GMRES method (denoted as GMRES(m)) is used in practice, since it constructs a Krylov subspace of dimension m , with m small at each cycle. However, if the appropriate m is not chosen, the convergence of the GMRES algorithm is not guaranteed, and the method may experience stagnation or slow convergence. Unfortunately, it is difficult to know how to choose m a priori and several researchers have proposed different adaptive choices of the parameter m . In this paper, we regard the GMRES(m) method as a control problem, in which the parameter m is the controlled variable and propose a new control-inspired strategy for choosing the parameter m adaptively at cycle j , using a discrete proportional-derivative controller which has the form:

$$m_{j+1} = m_j - \alpha \frac{\|r_j\|}{\|r_{j-1}\|} + \beta \frac{\|r_j\| - \|r_{j-2}\|}{2\|r_{j-1}\|},$$

where α and β are chosen experimentally. The advantage of this method, named PD-GMRES, is that only a few additional vectors need to be stored and the controller has the capacity to increase the dimension of the Krylov subspace if any convergence problem is detected.

Numerical comparisons between the proposed PD-GMRES method and the proposals of Baker [BJK09] and Gonalez [GC05] for choosing m were carried out. The test problems, chosen from the University of Florida’s Matrix Repository [bTDfUoFfAR], were: (1) add20, (2) cdde1, (3) circuit, (4) fpga_trans_01, (5) orsirr_1, (6) orsreg_1, (7) pde2961, (8) raefsky1, (9) raefsky2, (10) rdb2048, (11) sherman4, (12) steam2, (13) wang2, (14) watt_1, (15) young3c. For problems

with an unspecified right hand side, b was generated randomly using a uniform distribution with values between the minimum and maximum values in A . The numerical experiments carried out on this test set show that PD-GMRES has a better rate of convergence than GMRES(30), Baker's and Goncalvez' algorithms, for all but two (problems 7 and 10) of the 15 test problems. It is important to emphasize that the proposed PD-GMRES algorithm is easy to implement and showed good convergence on this test set.

Keywords: *Adaptive GMRES(m), proportional derivative (PD) controller, control formulation*

ÍNDICE GENERAL

LISTA DE FIGURAS	xiii
LISTA DE TABLAS	xvi
1 INTRODUCCIÓN	1
1.1 Originalidad y Relevancia	3
1.2 Organización del trabajo	4
2 GMRES Y GMRES(m)	5
2.1 Métodos de Proyección	5
2.2 GMRES	7
2.2.1 Por qué usar un subespacio de Krylov	8
2.2.2 Algoritmo	10
2.3 GMRES con reinicio: GMRES(m)	16
3 CONTROL Y GMRES(m)	19
3.1 Algunas definiciones	19
3.2 Control PID	21
3.3 Control y Algoritmos Iterativos	22
3.4 LQRES	24
3.5 GMRES(m) con m controlado	25
3.5.1 Medida de Convergencia	26
3.5.2 Justificación de la variación del parámetro de reinicio	27
3.5.3 Controlador Proporcional Derivativo PD(m)	27
3.5.4 Controlador Constante	30
3.6 GMRES(m) con m aleatorio	30
4 EXPERIMENTOS NUMÉRICOS	32
4.1 Resumen de los resultados	32
4.2 Reglas implementadas	36

4.2.1	PDRULE	36
4.2.2	Regla de Baker	37
4.2.3	Regla de Gonalez	38
4.3	Resultados por problema	39
4.3.1	Problema 1. Matriz Add20	39
4.3.2	Problema 2. Matriz Cdde1	41
4.3.3	Problema 3. Matriz Circuit_2	42
4.3.4	Problema 4. Matriz Fpga_trans_01	44
4.3.5	Problema 5. Matriz Orsirr_1	45
4.3.6	Problema 6. Matriz Orsreg_1	47
4.3.7	Problema 7. Matriz Pde2961	48
4.3.8	Problema 8. Matriz Raefsky1	50
4.3.9	Problema 9. Matriz Raefsky2	51
4.3.10	Problema 10. Matriz Rdb2048	53
4.3.11	Problema 11. Matriz Sherman4	54
4.3.12	Problema 12. Matriz Steam2	56
4.3.13	Problema 13. Matriz Wang2	57
4.3.14	Problema 14. Matriz Watt_1	59
4.3.15	Problema 15. Matriz Young3c	60
4.4	Consideraciones sobre estancamiento	62
4.4.1	Experimento de Convergencia Lenta	62
4.4.2	Experimentos de Estancamiento	64
5	CONCLUSIONES	68
5.1	Principales contribuciones	69
5.2	Trabajos futuros	69
ANEXO. PD-GMRES. EXPERIMENTOS PARA SELECCIONAR		
	α y γ	70
5.3	Elecci3n	71
5.4	Resumen de tiempos relativos	73
5.5	Tiempos de ejecuci3n de GMRES(30)	74
5.6	Problema 1. Matriz add20	74
5.7	Problema 2. Matriz Cdde1	75
5.8	Problema 3. Matriz Circuit_2	76
5.9	Problema 4. Matriz Fpga_trans_01	77
5.10	Problema 5. Matriz Orsirr_1	78
5.11	Problema 6. Matriz Orsreg_1	79

5.12 Problema 7. Matriz Pde2961	80
5.13 Problema 8. Matriz Raefsky1	81
5.14 Problema 9. Matriz Raefsky2	82
5.15 Problema 10. Matriz Rdb2048	83
5.16 Problema 11. Matriz Sherman4	84
5.17 Problema 12. Matriz Steam2	85
5.18 Problema 13. Matriz Wang2	86
5.19 Problema 14. Matriz Watt_1	87
5.20 Problema 15. Matriz Young3c	88

REFERENCIAS BIBLIOGRÁFICAS **90**

A APENDICE A. RESIDUOS PARALELOS Y MATRICES SIMÉTRICAS **94**

B APENDICE B. LQRES y LQRES(D) **98**

B.1 Control Optimo de Sistemas Lineales con costo cuadrático	98
B.2 Condiciones de estabilidad de un controlador LQR	104
B.3 LQRES y LQRES(D)	105
B.4 Experimentos	107
B.4.1 Experimentos con matriz de pequeñas dimensiones	107
B.4.2 Experimentos con matrices de problemas reales	108
B.5 Conclusión	111

LISTA DE FIGURAS

2.1	Relación de Arnoldi.	14
3.1	Sistema de Control Realimentado.	21
3.2	Diagrama de bloques para un algoritmo iterativo.	23
3.3	Diagrama de bloques para $GMRES(m)$	25
3.4	Diagrama de bloques para $GMRES(m)$ adaptativo.	26
3.5	Matriz Wang2: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	31
3.6	Matriz Wang2: Reinicios aleatorios m Vs Iteraciones.	31
4.1	Tiempos de ejecución relativos. Las barras están agrupadas por problemas. Cuanto más baja la altura de la barra, es mejor el tiempo que lleva hasta encontrar la solución. En el problema 15, solo PD-GMRES convergió. <i>Alg. 1</i> corresponde a la regla de Baker y <i>Alg. 2</i> a la regla de Gonzalez.	34
4.2	Estructura de Add20.	39
4.3	Matriz Add20: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	40
4.4	Matriz Add20: Reinicios m Vs Iteraciones.	40
4.5	Estructura de Cdde1.	41
4.6	Matriz Cdde1: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	41
4.7	Matriz Cdde1: Reinicios m Vs Iteraciones.	42
4.8	Estructura de Circuit_2.	42
4.9	Matriz Circuit2: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	43
4.10	Matriz Circuit2: Reinicios m Vs Iteraciones.	43
4.11	Estructura de Fpga_trans_01.	44
4.12	Matriz Fpga_trans_01: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	44
4.13	Matriz Fpga_trans_01: Reinicios m Vs Iteraciones.	45
4.14	Estructura de Orsirr_1.	45
4.15	Matriz Orsirr_1: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	46
4.16	Matriz Orsirr_1: Reinicios m Vs Iteraciones.	46

4.17	Estructura de Orsreg_1.	47
4.18	Matriz Orsreg_1: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	47
4.19	Matriz Orsreg_1: Reinicios m Vs Iteraciones.	48
4.20	Estructura de Pde2961.	48
4.21	Matriz Pde2961: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	49
4.22	Matriz Pde2961: Reinicios m Vs Iteraciones.	49
4.23	Estructura de Raefsky1.	50
4.24	Matriz Raefsky1: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	50
4.25	Matriz Raefsky1: Reinicios m Vs Iteraciones.	51
4.26	Estructura de Raefsky2.	51
4.27	Matriz Raefsky2: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	52
4.28	Matriz Raefsky2: Reinicios m Vs Iteraciones.	52
4.29	Estructura de Rdb2048.	53
4.30	Matriz Rdb2048: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	53
4.31	Matriz Rdb2048: Reinicios m Vs Iteraciones.	54
4.32	Estructura de Sherman4.	54
4.33	Matriz Sherman4: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	55
4.34	Matriz Sherman4: Reinicios m Vs Iteraciones.	55
4.35	Estructura de Steam2.	56
4.36	Matriz Steam2: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	56
4.37	Matriz Steam2: Reinicios m Vs Iteraciones.	57
4.38	Estructura de Wang2.	57
4.39	Matriz Wang2: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	58
4.40	Matriz Wang2: Reinicios m Vs Iteraciones.	58
4.41	Estructura de Watt_1.	59
4.42	Matriz Watt_1: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	59
4.43	Matriz Watt_1: Reinicios m Vs Iteraciones.	60
4.44	Estructura de Young3c.	60
4.45	Matriz Young3c: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	61
4.46	Matriz Young3c: Reinicios m Vs Iteraciones.	61
4.47	Matriz Young3c: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	63
4.48	Matriz Young3c: Reinicios m Vs Iteraciones.	63
4.49	Estructura de Mahindas.	64
4.50	Matriz Mahindas: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	65
4.51	Matriz Mahindas: Reinicios m Vs Iteraciones.	65
4.52	Estructura de Shl_400.	66
4.53	Matriz Shl_400: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	66

4.54	Matriz Shl_400: Reinicios m Vs Iteraciones.	67
A.1	$r_i - AV_my \perp AV_my$, where $r_{i+1} = r_i - AV_my$	95
B.1	Sistema realimentado LQR de horizonte finito (Linear quadratic regulator).	103
B.2	Residuo como Sistema realimentado LQR en estado estable.	106
B.3	LQRES: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	108
B.4	LQRES y LQRES(D): $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	108
B.5	LQRES: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	109
B.6	LQRES(D): $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	109
B.7	LQRES: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	110
B.8	LQRES(D): $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	110
B.9	LQRES: $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	111
B.10	LQRES(D): $\log(\ r_j\ /\ r_0\)$ Vs Iteraciones	111

LISTA DE TABLAS

4.1	Lista de problemas. n es la dimensión de la matriz y nnz es el número de coeficientes distintos de cero	33
5.1	Tiempos por problema y algoritmo. $\alpha = 3$, $\gamma = -9$, 5 mediciones de tiempo.	72
5.2	Tiempos por problema y algoritmo. $\alpha = 3$, $\gamma = -5$, 5 mediciones de tiempo.	72
5.3	Sumatoria de tiempos relativos. $tabla(\alpha, \gamma) = \sum_{prob=1}^{15} \frac{t(PD-GMRES)}{t(GMRES(30))}$	73
5.4	Menores tiempos relativos. $tiempo(\alpha, \gamma) = \sum_{prob=1}^{15} \frac{t(PD-GMRES)}{t(GMRES(30))}$	73
5.5	Tiempo de ejecución de $GMRES(30)$ en segundos	74
5.6	add20.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	74
5.7	add20.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	75
5.8	cdde1.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	75
5.9	cdde1.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	76
5.10	circuit_2.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	76
5.11	circuit_2.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	77
5.12	fpga_trans_01.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	77
5.13	fpga_trans_01.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	78
5.14	orsirr_1.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	78
5.15	orsirr_1.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	79
5.16	orsreg_1.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	79
5.17	orsreg_1.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	80
5.18	pde2961.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	80
5.19	pde2961.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	81
5.20	raefsky1.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	81
5.21	raefsky1.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	82
5.22	raefsky2.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	82
5.23	raefsky2.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	83

5.24	rdb2048.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	83
5.25	rdb2048.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	84
5.26	sherman4.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	84
5.27	sherman4.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	85
5.28	steam2.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	85
5.29	steam2.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	86
5.30	wang2.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	86
5.31	wang2.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	87
5.32	watt_1.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	87
5.33	watt_1.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	88
5.34	young3c.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos	88
5.35	young3c.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos	89

Capítulo 1

INTRODUCCIÓN

Un sistema de ecuaciones lineales

$$Ax = b \tag{1.1}$$

donde $A \in \mathbb{F}^{n \times n}$ y $\mathbb{F} = \mathbb{C}$ o $\mathbb{F} = \mathbb{R}$, puede ser resuelto utilizando métodos directos o métodos iterativos. Los métodos directos calculan la solución exacta (suponiendo aritmética de precisión infinita). Un ejemplo de este tipo de métodos es la eliminación gaussiana o factorización LU . El problema con los métodos directos clásicos es que no aprovechan la estructura de la matriz A y en general su complejidad temporal es del orden de $O(n^3)$. Por otra parte, los métodos iterativos calculan una solución aproximada, aprovechan la estructura u otras propiedades de la matriz A , y dadas ciertas condiciones consiguen calcular la solución con una complejidad de $O(n^2)$ [TB97].

Un método iterativo construye aproximaciones sucesivas a la solución de (1.1). En la iteración k , la aproximación es x_k y el residuo $r_k = b - Ax_k$. Sea x^* la solución exacta de (1.1). El algoritmo iterativo converge idealmente cuando $\exists k$, $x_k = x^*$. En la práctica se utiliza una medida de convergencia, por ejemplo se considera que el algoritmo convergió cuando $\|r_k\|/\|r_0\| < \textit{tolerancia}$.

Los métodos iterativos explotan propiedades de la matriz A para conseguir un orden de complejidad bajo. Entre esas propiedades podemos citar la esparcidad (si tiene muchos ceros), la simetría y la distribución de ceros que tiene. No hay un método iterativo que sea el mejor para cualquier matriz.

En esta tesis partimos del método iterativo de Residuo Mínimo Generalizado (GMRES) usado normalmente para la resolución de sistemas de ecuaciones lineales (1.1), cuando la matriz A es no-hermitiana o no-simétrica. GMRES, propuesto por Saad y Schultz en 1986 [SS86], parte de una aproximación inicial x_0 (de su residuo respectivo $r_0 = b - Ax_0$), y aproxima la solución de (1.1) con un vector x_k ,

tal que la norma l_2 de su residuo es mínimo. La aproximación x_k pertenece a un subespacio de Krylov. El subespacio de Krylov de orden k correspondiente a la matriz A y al residuo r_0 es:

$$K_k(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0\}. \quad (1.2)$$

GMRES construye una base ortogonal de dimensión k para $K_k(A, r_0)$ utilizando el método de Arnoldi. Este método es un método general para matrices no-hermitianas que reduce la matriz A a una matriz de Hessenberg superior [and97, Saa03]. La ortogonalización puede ser implementada usando el método Gram-Schmidt modificado o el método de Householder. Aunque el método de Householder es más estable, es más lento que el método Gram-Schmidt modificado [Saa03]. Luego de construir la base ortogonal, GMRES resuelve un problema de mínimos cuadrados para minimizar la norma del residuo r_k y obtener la aproximación x_k . Si el orden k del subespacio de Krylov es igual a la dimensión n de la matriz A , GMRES encuentra la solución exacta de (1.1). Usando MGS (Gram-Schmidt modificado) para la construcción de una base ortogonal de dimensión k , suponiendo que la multiplicación de matriz-vector tiene una complejidad menor que $O(n^2)$, la complejidad temporal de GMRES es $O(k^2n)$

Sin embargo, al usar k cercano a n , sus necesidades de memoria y cálculo lo hacen muy costoso. De hecho, si hacemos $k = n$, su complejidad temporal pasa a ser $O(n^3)$. Por ese motivo, Saad y Schultz [SS86] propusieron utilizar GMRES con reinicio, conocido en la literatura como Restarted GMRES o GMRES(m). En cada iteración j de GMRES(m), GMRES calcula una aproximación x_j usando un subespacio $K_m(A, r_{j-1})$. Luego, realiza un test de convergencia. Si no convergió, la siguiente iteración usa x_j como x inicial para una nueva llamada a GMRES. Si convergió, termina y x_j es la solución buscada. La aproximación inicial x_0 es un dato, generalmente se usa $x_0 = 0$. La ventaja de usar el método con reinicio es que como máximo se realizan m iteraciones del proceso de Arnoldi (m ortogonalizaciones), por eso los costos de memoria y cálculo están bajo control.

En general, se utiliza GMRES(m) con un parámetro $m \ll n$ y constante. En casos prácticos se suele utilizar $m = 30$, por ejemplo PETSC [BBG⁺09] usa este valor de forma predeterminada. Broyden y Vespucci utilizan los valores $m = 10, 20, 30$ para sus experimentos [CB04].

El problema de GMRES(m) es que su convergencia no está garantizada [Emb03]. De hecho, si el parámetro m no es elegido de forma apropiada el método puede experimentar una convergencia lenta o un estancamiento de la solución ($r_{j+1} \approx r_j$ para todo j). Si GMRES(m) se estanca, una estrategia podría ser aumentar

m . Sin embargo, esto no siempre garantiza una convergencia más rápida y en algunos casos tener un m mayor puede ser prohibitivo [EES00, Emb03, SS07]. Desafortunadamente, es difícil saber como elegir m a priori.

Varios trabajos han propuesto diferentes alternativas para seleccionar el parámetro m de forma adaptativa. Estas estrategias pueden ser categorizadas en tres grupos:

- reglas simples, con parámetros elegidos empíricamente, pero no tan simples de estimar [SWK98, HN03].
- reglas que incluyen cálculos no triviales de autovalores o ceros de polinomios [MN00, ZN05].
- reglas empíricas simples [BJK09, GC05].

En esta tesis, buscamos una forma de variar el parámetro de reinicio de GMRES(m) para tratar de mejorar su convergencia. El objetivo consiste en aplicar la Teoría de Control para encontrar esa estrategia. Para conseguir esto, formulamos GMRES(m) como un problema de control, en el que la variable controlada es m y proponemos una estrategia nueva para elegir el parámetro m de forma adaptativa en cada iteración j . Este método tiene la ventaja de que necesita almacenar pocos vectores adicionales, y el controlador altera (incrementa o disminuye) la dimensión del subespacio de Krylov si detecta problemas. Debemos mencionar que Helmke et al. [HJL06] han modelado GMRES(m) como un caso de los métodos de Runge-Kutta. También han propuesto un nuevo algoritmo LQRES que necesita la solución de una ecuación algebraica de Riccati. La solución de esta ecuación es intratable para matrices con dimensión mayor o igual que 1000.

El resultado principal de este trabajo es el uso de una técnica inspirada en un controlador PD (Proporcional Derivativo), que en la mayoría de los problemas utilizados tuvo mejores resultados que técnicas encontradas en la literatura. Los problemas de prueba corresponden a matrices disponibles en [bTDfUoFfAR].

El método de verificación utilizado es experimental. De forma a verificar la eficiencia de la estrategia de control propuesta y del GMRES(m) resultante, comparamos los resultados con reglas de elección de m propuestas por Baker et al. [BJK09] y por González et al. [GC05].

1.1 Originalidad y Relevancia

Las técnicas de control han sido utilizadas para la resolución de sistemas del tipo (1.1) en [BK04]. La presente tesis implementa el concepto de la teoría de control

en el contexto de GMRES y presenta los siguientes avances:

- Formula GMRES(m) como un sistema de control (§3.5).
- Presenta una estrategia nueva para controlar el parámetro de reinicio m de GMRES(m) inspirada den la teoría de Control (§3.5.3,§4.2.1).
- Interpreta el método de Baker et al. presentado en [BJK09] como una forma de controlar el parámetro de reinicio m (§3.5.4).

1.2 Organización del trabajo

En el Capítulo 2 presentamos los métodos iterativos GMRES y GMRES(m) con sus características resaltantes. En el Capítulo 3, presentamos las estrategia propuesta (§3.5.3), también interpretamos la estrategia de Baker desde el punto de vista de Control (§3.5.4). Los resultados experimentales son presentados en el Capítulo 4, las matrices utilizadas son matrices no-simétricas obtenidas de la Colección de Matrices Dispersas (poco densas) de la Universidad de Florida [bTDfUoFfAR]. Esta colección incluye matrices del Mercado de Matrices [MfNU]. En este capítulo también incluimos un problema con convergencia lenta y dos con estancamiento. Finalmente, presentamos las conclusiones y trabajos futuros en el Capítulo 5.

Capítulo 2

GMRES Y GMRES(m)

El Método de Residuo Mínimo Generalizado (GMRES), que fue presentado por Saad y Schultz en 1986 [SS86], es ampliamente utilizado para la resolución de sistemas de ecuaciones lineales no simétricos, generalmente poco densos. Es un algoritmo iterativo de proyección que utiliza subespacios de Krylov. La estabilidad en retroceso del algoritmo que utiliza Modified Gram-Schmidt fue demostrada en 2006 por Paige et al. [PRS06]. Es un método robusto. Aún cuando hay errores al hacer operaciones de multiplicación de matriz por vector, da buenas soluciones [SS03, vdES04].

Sin embargo, para matrices de gran porte, los requerimientos computacionales hacen que GMRES sea costoso cuando el número de pasos es cercano a n . Como una forma de disminuir este costo, Saad y Schultz sugieren el uso de GMRES con reinicio, generalmente denominado GMRES(m) [SS86]. En este capítulo primero describimos los métodos de proyección, luego presentamos los algoritmos GMRES y GMRES(m).

Aunque trabajamos con matrices reales, GMRES y GMRES(m) también funcionan con matrices complejas [Saa03].

2.1 Métodos de Proyección

Dado el sistema de ecuaciones

$$Ax = b, \tag{2.1}$$

donde A es una matriz $n \times n$, las técnicas de proyección buscan una solución en un subespacio de \mathbb{R}^n . Sea \mathcal{K} este subespacio de búsqueda que contiene a las soluciones aproximadas. Si su dimensión es m , se imponen m restricciones para calcular la aproximación. Comúnmente se utilizan m restricciones de ortogonalidad que son linealmente independientes. En general, se hace que el residuo $r = b - Ax$ sea

ortogonal a m vectores linealmente independientes. Esto define el subespacio de restricciones \mathcal{L} . Esta forma de resolver el problema es común a muchos métodos y las m restricciones se conocen como condiciones de Petrov-Galerkin.[Saa03]

Los métodos de proyección en los que $\mathcal{K} = \mathcal{L}$ se llaman de proyección ortogonal. Cuando \mathcal{L} es diferente de \mathcal{K} , se llaman de proyección oblicua [Saa03]. En términos de fórmulas, dada una aproximación inicial x_0 el proceso de proyección busca una solución \tilde{x} tal que

$$\begin{aligned} \tilde{x} &\in x_0 + \mathcal{K}, \\ r &= (b - A\tilde{x}) \perp \mathcal{L} \end{aligned} \quad (2.2)$$

Escribiendo $\tilde{x} = x_0 + \delta$, el residuo inicial $r_0 = b - Ax_0$ y el producto interior $\langle \cdot, \cdot \rangle$, la condición de ortogonalidad $r = (r_0 - A\delta) \perp \mathcal{L}$ es

$$\langle r_0 - A\delta, w \rangle = 0 \quad \forall w \in \mathcal{L} \quad (2.3)$$

Sea $V = [v_1, v_2, \dots, v_m]$ una matriz cuyas columnas son una base del subespacio \mathcal{K} y $W = [w_1, w_2, \dots, w_m]$ una matriz cuyas columnas son una base del subespacio \mathcal{L} , entonces podemos escribir $\tilde{x} = x_0 + Vy$ y por la condición de ortogonalidad del residuo (2.3) obtenemos $\langle r_0 - AVy, w_i \rangle = 0$.

Note que si se cumple la condición de ortogonalidad se cumple para la base de \mathcal{L} , se cumple $\forall w \in \mathcal{L}$, por lo que podemos escribir

$$w^T(r_0 - AVy) = 0 \quad (2.4)$$

Usando (2.4) y la matriz W tenemos

$$W^T r_0 = W^T AVy \quad (2.5)$$

donde la incógnita es y . Resolviendo 2.5 obtenemos

$$y = (W^T AV)^{-1} W^T r_0 \quad (2.6)$$

De lo anterior, podemos escribir un algoritmo general para un método de proyección.

mientras (*no converja*) **hacer**

Elegir los subespacios \mathcal{K} y \mathcal{L} ;

Elegir las bases respectivas $V = [v_1, v_2, \dots, v_m]$ y $W = [w_1, w_2, \dots, w_m]$

;

$r = b - Ax$;

$y = (W^T AV)^{-1} W^T r$;

$x = x + Vy$;

fin_mientras

Algoritmo 1: Algoritmo General de Proyección

Con una secuencia adecuada de subespacios \mathcal{K} y \mathcal{L} , el Algoritmo (1) converge a la solución del sistema. Notar que si la matriz A es no-singular, entonces la matriz $W^T AV$ también. [Saa03]

Lema 1 ([Saa03]). Sean A , \mathcal{K} y \mathcal{L} tales que satisfacen una de las condiciones:

(i) A es definida positiva y $\mathcal{L} = \mathcal{K}$

(ii) A es no singular y $\mathcal{L} = A\mathcal{K}$

Entonces, la matriz $B = W^T AV$ es no singular para toda base V , W de \mathcal{K} y \mathcal{L} respectivamente.

Demostración. Primero consideramos el caso (i). Sea V cualquier base de \mathcal{K} y W cualquier base de \mathcal{L} . Como $\mathcal{K} = \mathcal{L}$, podemos escribir $W = VG$ donde G es una matriz no-singular. Reescribimos $B = W^T AV = G^T V^T AV$. Dado que G es no-singular, y A definida positiva, entonces B es no-singular.

Para el caso (ii). Sea V cualquier base de \mathcal{K} y W cualquier base de \mathcal{L} . Como $\mathcal{L} = A\mathcal{K}$, podemos escribir $W = AVG$ donde G es una matriz no-singular. Reescribimos $B = W^T AV = G^T V^T A^T AV = G^T (AV)^T AV$. Como A es no-singular, AV tiene rango completo, es decir es no-singular, por lo que B es no-singular.

□

2.2 GMRES

En esta sección consideramos el método GMRES como un caso particular de los métodos de proyección donde $\mathcal{K} = \mathcal{K}_k(A, r_0) = \{r_0, Ar_0, A^2 r_0, \dots, A^{k-1} r_0\}$ es un subespacio de Krylov, y $\mathcal{L} = A\mathcal{K}$.

2.2.1 Por qué usar un subespacio de Krylov

La convergencia de los métodos que utilizan subespacios de Krylov está relacionada con el grado del polinomio mínimo de la matriz A . El polinomio mínimo $q(t)$ de A es un polinomio mónico con grado mínimo y se construye a partir de los autovalores de A . [IM98]

Sea A una matriz de dimensión $n \times n$ y sus autovalores no repetidos $\lambda_1, \dots, \lambda_k$. Sea m_j el índice de λ_j (el tamaño del mayor bloque de Jordan asociado a λ_j). Su polinomio mínimo es

$$q(t) = \prod_{j=1}^k (t - \lambda_j)^{m_j} \quad (2.7)$$

y su grado correspondiente es $m = \sum_{j=1}^k m_j$.

Efectuando los productos de (2.7) y usando los coeficientes α_j obtenidos, podemos escribir

$$q(t) = \sum_{j=0}^m \alpha_j t^j \quad (2.8)$$

Por el teorema de Caley-Hamilton, A satisface $q(A) = 0$, usando (2.8) y suponiendo A no-singular, tenemos

$$\begin{aligned} -\alpha_0 I &= \alpha_1 A + \dots + \alpha_m A^m \\ -\alpha_0 I &= A(\alpha_1 + \dots + \alpha_m A^{m-1}) \\ -A^{-1} \alpha_0 I &= A^{-1} A(\alpha_1 + \dots + \alpha_m A^{m-1}) \\ A^{-1} &= -\frac{1}{\alpha_0} \sum_{j=0}^{m-1} \alpha_{j+1} A^j \end{aligned} \quad (2.9)$$

Multiplicando (2.9) por b a la izquierda obtenemos $A^{-1}b = -\frac{1}{\alpha_0} \sum_{j=0}^{m-1} \alpha_{j+1} A^j b$, recordando que $x = A^{-1}b$ vemos que $x \in K_m(A, b)$. Consecuentemente, cuando menor sea el grado del polinomio mínimo de la matriz A , más rápidamente encontraremos la aproximación de la solución x^* . [Saa03, IM98]

Si A es no-singular, la solución puede estar o no en un subespacio de Krylov. Si queremos que la solución de un sistema general pertenezca a un subespacio de Krylov, debemos restringir b alejándolo de la parte nilpotente de A [IM98]. La descomposición de Jordan descompone una matriz A en $A = D + N$ donde D es la parte diagonalizable o semisimple y N es la parte nilpotente. Para ver que debemos alejar b de N , supongamos que $Nx = c$ es un sistema consistente donde N es nilpotente y $c \neq 0$. Como N es nilpotente $\exists i$ tal que $N^i = 0$ y $N^{i-1} \neq 0$. Supongamos que una solución x pertenece al subespacio de Krylov $\mathcal{K}(N, c)$, es

decir $x = \xi_0 c + \xi_1 cN + \cdots + \xi_{i-1} N^{i-1} c$. Como x es solución, tenemos

$$\begin{aligned} N(\xi_0 c + \xi_1 cN + \cdots + \xi_{i-1} N^{i-1} c) &= c \\ (I - N\xi_0 - \xi_1 N^2 + \cdots - \xi_{i-2} N^{i-1})c &= 0 \end{aligned}$$

Por un lado suponemos $c \neq 0$. Por otro lado, suponemos que el sistema $Nx = c$ es consistente, por lo que $(I - N\xi_0 + \xi_1 N^2 + \cdots + \xi_{i-2} N^{i-1}) \neq 0$. Esto es una contradicción con el resultado anterior. Entonces, el sistema consistente $Nx = c$ con N nilpotente no tiene solución en un subespacio de Krylov. [IM98]

Teorema 2 ([IM98] Existencia de una solución en un subespacio de Krylov). *Un sistema lineal $Ax = b$ tiene una solución en un subespacio de Krylov si y solo si $b \in \mathcal{R}(A^i)$, donde i es el índice del autovalor 0 de A . Es decir, i es el tamaño del mayor bloque de Jordan asociado al autovalor 0.*

Si A es no-singular, no tiene 0 entre sus autovalores. Es decir, su descomposición en la forma de Jordan no tiene parte nilpotente. Como vimos antes, la solución pertenece a un subespacio de Krylov. Si A es singular con autovalor 0 con índice o multiplicidad 1, la condición requiere que $b \in \mathcal{R}(A)$. Sea d el número de autovalores distintos de A y $Ax = b$ consistente. Si A es singular, existe una solución $x \in K_{d-1}(A, b)$. Si A es no-singular, existe la solución $x \in K_d(A, b)$.

Para el caso no-singular vimos que A^{-1} se puede representar usando un subespacio de Krylov, por eso es natural buscar una solución en ese subespacio. Por otra parte, cuando A es singular no hay una inversa. [IM98] utiliza la pseudoinversa de Drazin y demuestra que hay una solución en el subespacio de Krylov.

Sea la matriz A con autovalor 0 de índice i , la pseudoinversa de Drazin A^D cumple las condiciones a) $A^D A A^D = A^D$, b) $A^D A = A A^D$ y c) $A^{i+1} A^D = A^i$.

Teorema 3 ([IM98] Unicidad de la solución que pertenece a un subespacio de Krylov). *Sea m el grado del polinomio mínimo de A , y sea i el índice del autovalor 0 de A . Si $b \in \mathcal{R}(A^i)$ ($\mathcal{R}(A^i)$ rango de la parte no-singular de A), el sistema lineal $Ax = b$ tiene una única solución que pertenece a un subespacio de Krylov. $x = A^D b \in \mathcal{K}_{m-i}(A, b)$. Si $b \notin \mathcal{R}(A^i)$, entonces $Ax = b$ no tiene solución en el subespacio de Krylov $\mathcal{K}_m(A, b)$.*

En resumen:

1. Si A es no singular, la solución pertenece al subespacio de Krylov.
2. Si A es singular y se cumplen las condiciones del Teorema anterior, hay una solución que pertenece al subespacio de Krylov.

Por lo tanto, un subespacio de Krylov es una buena opción para buscar una solución para el sistema $Ax = b$.

2.2.2 Algoritmo

El Método de Residuo Mínimo Generalizado (GMRES) fue presentado por Saad y Schultz en 1986 [SS86]. Sea x_0 una aproximación inicial de la solución. Sea $r_0 = b - Ax_0$ el residuo inicial y $\mathcal{K}_k(A, r_0) = \{r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0\}$ el subespacio de Krylov correspondiente a A y r_0 . En el k -ésimo paso, GMRES busca una solución de la forma

$$x_k = x_0 + V_k y_k \quad (2.10)$$

Donde V_k es una matriz, en general densa, V_k con dimensión $n \times k$ elementos. Sus columnas forman una base ortogonal de $\mathcal{K}_k(A, r_0)$. $y_k \in \mathbb{R}^k$ es tal que minimiza la norma l_2 (norma Euclidiana) del residuo:

$$\|r_k\| = \|b - Ax_k\|_2 = \min_{x \in x_0 + \mathcal{K}_k(A, r_0)} \|b - Ax\|_2. \quad (2.11)$$

Usando la relación de Arnoldi $AV_k = V_{k+1}H_k$ el lado izquierdo de la expresión (2.11) toma la forma:

$$\|r_0 - AV_k y_k\| = \|r_0 - V_{k+1}H_k V_k y_k\| \quad (2.12)$$

$$= \|V_{k+1}\| \|\beta e_1 - H_k y_k\| \quad (2.13)$$

$$= \|\beta e_1 - H_k y_k\| \quad (2.14)$$

donde V_{k+1} es una matriz ortogonal y $\beta = \|r_0\|$. Es decir, GMRES tiene como

entrada r_0 y retorna $z_k = V_k y_k$, la solución x se obtiene usando (2.10). [SS86]

Entrada: A matriz del sistema, $r_0 = b - Ax_0$ residuo inicial, m dimensión del subespacio de Krylov

Salida: $z = V_{:,1:i} y$ a ser usado para calcular la aproximación $x = x_0 + z$

$\beta = \|r_0\|_2$

$v_{:,1} = r_0/\beta$

desde $i \leftarrow 1, m$ **hacer**

$z = Av_{:,i}$

desde $j = 1, \dots, i$ **hacer**

$h_{j,i} = (v_{:,j})^T z$

$z = z - h_{j,i} v_{:,j}$

fin_desde

$h_{i+1,i} = \|z\|$

si $h_{i+1,i} = 0$ **entonces**

break

fin_si

$v_{:,i+1} = z/h_{i+1,i}$

$\min \|\beta e_1 - H_{1:(i+1),1:i} y\|_2$ for y

salir Si el residuo cumple tolerancia indicada

fin_desde

$z = V_{:,1:i} y$

retornar: z

Algoritmo 2: GMRES

El algoritmo de Arnoldi puede verse como una forma de extraer información de una matriz A cuando solo se dispone del producto de A por un vector [Dem97]. Calculamos la secuencia $y_1 = b, y_2 = Ay_1, y_3 = Ay_2 = A^2 y_1, \dots, y_n = A^{n-1} y_1$. Construimos la matriz K que tiene como columnas las y_i , es decir $K = [y_1 y_2, \dots, y_n]$. Al multiplicar K por A obtenemos $AK = [Ay_1 Ay_2 \dots Ay_n] = [y_2 y_3 \dots y_n A^n y_1]$. Suponemos que K es no-singular. Sea e_i el i -ésimo vector base de la base canónica de \mathcal{R}^n y la constante $c = K^{-1} A^n y_1$. Con esto podemos escribir

$$AK = K[e_2 e_3 \dots e_n - c] \quad (2.15)$$

Multiplicando (2.15) por K^{-1} por la izquierda, tenemos

$$K^{-1}AK = C \quad (2.16)$$

En (2.16), la matriz C es

$$C = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 & -c_1 \\ 1 & 0 & & 0 & 0 & -c_2 \\ 0 & 1 & & & & \\ & 0 & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & \dots & 1 & 0 & \\ & & \dots & & 1 & -c_n \end{bmatrix} \quad (2.17)$$

Es importante notar que la matriz C es Hessenberg superior. Es decir, tiene ceros por debajo de la subdiagonal. La transformación (2.16), aunque interesante, no se usa ya que aún cuando A sea dispersa o poco densa, K es densa. Además, es más probable que K esté mal condicionada, dado que y_i converge al autovalor de mayor valor de A , por lo que las columnas de K se hacen cada vez más paralelas. Usando una descomposición QR , factorizamos $K = VR$, donde V es una matriz ortonormal y R una matriz triangular superior. Llevando $K = VR$ en (2.16)

$$\begin{aligned} (VR)^{-1}A(VR) &= C, \\ R^{-1}V^TAVR &= C, \\ V^TAV &= RCR^{-1}, \\ V^TAV &= H, \\ AV &= V^TH. \end{aligned} \quad (2.18)$$

donde (2.18) es una relación en la que todas las matrices son de dimensión $n \times n$. $H = RCR^{-1}$ es una matriz Hessenberg porque C es Hessenberg y R es triangular superior.

Por otra parte, si calculamos la columna j en (2.18) tenemos $Av_j = Vh_j$. Dado que el lado derecho es una combinación lineal de las columnas de V , $Vh_j = \sum_{i=1}^n h_{i,j}v_i$. Pero como H es Hessenberg, podemos escribir $Vh_j = \sum_{i=1}^{j+1} h_{i,j}v_i$. Es decir,

$$Av_j = \sum_{i=1}^{j+1} h_{i,j}v_i. \quad (2.19)$$

Para obtener el valor de $h_{k,i}$ con $1 \leq k \leq j$ multiplicamos (2.19) por la izquierda

por v_k^T . Dado que las v_i son ortonormales, tenemos

$$\begin{aligned} v_k^T Av_j &= h_{k,j} v_k^T v_k, \\ v_k^T Av_j &= h_{k,j}, \\ \langle Av_j, v_k \rangle &= h_{k,j}. \end{aligned} \quad (2.20)$$

Para obtener el valor de $h_{j+1,j}$ usamos nuevamente (2.19). Esta vez despejamos $h_{j+1,j}v_j$

$$h_{j+1,j}v_j = Av_j - \sum_{i=1}^j h_{i,j}v_i. \quad (2.21)$$

Calculando la norma l_2 a ambos lados de (2.21), y usando la ortonormalidad de v_j

$$\begin{aligned} h_{j+1,j}^2 \|v_j\|^2 &= \|Av_j - \sum_{i=1}^j h_{i,j}v_i\|^2, \\ h_{j+1,j} &= \|Av_j - \sum_{i=1}^j h_{i,j}v_i\|. \end{aligned} \quad (2.22)$$

Llevando el resultado de (2.22) en (2.21) obtenemos:

$$v_{j+1,j} = Av_j - \sum_{i=1}^j h_{i,j}v_i / \|Av_j - \sum_{i=1}^j h_{i,j}v_i\| \quad (2.23)$$

Usando (2.20), (2.22) y (2.23), tenemos los cálculos que hace el Algoritmo (2). [Dem97]

Cálculo parcial de la relación de Arnoldi

Si consideramos un cálculo parcial de V hasta la columna m en (2.19), obtenemos el mayor valor de $j = m$. Usamos este valor en el lado derecho y obtenemos el mayor valor de $i = m + 1$. Por lo tanto se cumple

$$A_{n \times n} V_{n \times m} = V_{n \times m+1} \tilde{H}_{m+1 \times m} \quad (2.24)$$

Como ejemplo consideramos la matriz A de dimensión 5×5 y V de dimensión 5×3 .

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} \\ a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} & a_{5,5} \end{bmatrix} \begin{bmatrix} v_{1,1} & v_{1,2} & v_{1,3} \\ v_{2,1} & v_{2,2} & v_{2,3} \\ v_{3,1} & v_{3,2} & v_{3,3} \\ v_{4,1} & v_{4,2} & v_{4,3} \\ v_{5,1} & v_{5,2} & v_{5,3} \end{bmatrix} = \begin{bmatrix} v_{1,1} & v_{1,2} & v_{1,3} & v_{1,4} \\ v_{2,1} & v_{2,2} & v_{2,3} & v_{2,4} \\ v_{3,1} & v_{3,2} & v_{3,3} & v_{3,4} \\ v_{4,1} & v_{4,2} & v_{4,3} & v_{4,4} \\ v_{5,1} & v_{5,2} & v_{5,3} & v_{5,4} \end{bmatrix} \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ 0 & h_{3,2} & h_{3,3} \\ 0 & 0 & h_{4,3} \end{bmatrix}$$

□

Buscando tener $V_{n \times m}$ en el lado derecho de (2.24), analizamos la matriz $\tilde{H}_{m+1 \times m}$. En su fila $m+1$ solo el elemento $h_{m+1 \times m}$ es distinto de cero por lo que la diferencia está en el cálculo de la columna m del producto $V_{n \times m+1} \tilde{H}_{m+1 \times m}$. A cada fila i de la columna m le falta el sumando a $v_{i,m+1} h_{m+1,m}$. Por este motivo podemos escribir

$$A_{n \times n} V_{n \times m} = V_{n \times m} H_{m \times m} + w e_m^T \quad (2.25)$$

Donde H corresponde a la matriz \tilde{H} sin su última fila, e_m es la base canónica m de F^m y w es el vector

$$w = \begin{bmatrix} v_{1,m+1} h_{m+1,m} \\ v_{2,m+1} h_{m+1,m} \\ \vdots \\ v_{n,m+1} h_{m+1,m} \end{bmatrix} = h_{m+1,m} \begin{bmatrix} v_{1,m+1} \\ v_{2,m+1} \\ \vdots \\ v_{n,m+1} \end{bmatrix} \quad (2.26)$$

Vemos esta relación de forma gráfica en la Figura 2.1.

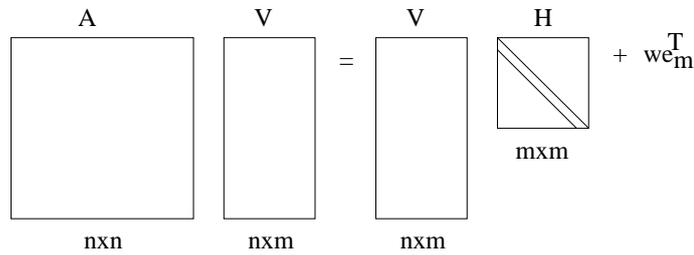


Figura 2.1: Relación de Arnoldi.

Un ejemplo de (2.25) es:

$$\begin{aligned}
 \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} \\ a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} & a_{5,5} \end{bmatrix} \begin{bmatrix} v_{1,1} & v_{1,2} & v_{1,3} \\ v_{2,1} & v_{2,2} & v_{2,3} \\ v_{3,1} & v_{3,2} & v_{3,3} \\ v_{4,1} & v_{4,2} & v_{4,3} \\ v_{5,1} & v_{5,2} & v_{5,3} \end{bmatrix} &= \begin{bmatrix} v_{1,1} & v_{1,2} & v_{1,3} \\ v_{2,1} & v_{2,2} & v_{2,3} \\ v_{3,1} & v_{3,2} & v_{3,3} \\ v_{4,1} & v_{4,2} & v_{4,3} \\ v_{5,1} & v_{5,2} & v_{5,3} \end{bmatrix} \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ 0 & h_{3,2} & h_{3,3} \end{bmatrix} \\
 &+ h_{4,3} \begin{bmatrix} 0 & 0 & v_{1,4} \\ 0 & 0 & v_{2,4} \\ 0 & 0 & v_{3,4} \\ 0 & 0 & v_{4,4} \\ 0 & 0 & v_{5,4} \end{bmatrix} \tag{2.27}
 \end{aligned}$$

donde

$$h_{4,3} \begin{bmatrix} 0 & 0 & v_{1,4} \\ 0 & 0 & v_{2,4} \\ 0 & 0 & v_{3,4} \\ 0 & 0 & v_{4,4} \\ 0 & 0 & v_{5,4} \end{bmatrix} = h_{4,3} \begin{bmatrix} v_{1,4} \\ v_{2,4} \\ v_{3,4} \\ v_{4,4} \\ v_{5,4} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

□

Si multiplicamos ambos lados de (2.25) por V^T que es la inversa de V , tenemos:

$$\begin{aligned}
 V_{n \times m}^T A_{n \times n} V_{n \times m} &= V_{n \times m}^T V_{n \times m} H_{m \times m} + V_{n \times m}^T w e_m^T \\
 V_{n \times m}^T A_{n \times n} V_{n \times m} &= H_{m \times m} \tag{2.28}
 \end{aligned}$$

Donde el término $V_{n \times m}^T w e_m^T$ se anula porque las columnas de V forman una base orthonormal.

Los resultados anteriores prueban la siguiente proposición:

Proposición 2.2.1 ([Saa03]). *Sea V_m la matriz $n \times m$ con los vectores columna v_1, v_2, \dots, v_m . Sea \tilde{H}_m la matriz $(m+1) \times m$ la matriz de Hessenberg cuyos elementos $h_{i,j}$ son calculados por el algoritmo de Arnoldi. Sea H_m la matriz $m \times m$ obtenida al borrar la fila $m+1$ de \tilde{H}_m y e_m la m -ésima base canónica.*

Entonces se cumplen las siguientes relaciones:

- (i) $AV_m = V_m H_m + w e_m^T$
- (ii) $AV_m = V_{m+1} \tilde{H}_m$
- (iii) $V_m^T AV_m = H_m$

Si $h_{i+1,i} = 0$, v_{i+1} no puede ser calculado, entonces $AV_i = V_i H$. Recordando que GMRES minimiza la norma del residuo $b - A(x_0 + V_i y)$, tenemos

$$\|r_0 - AV_i y\| = \|\beta v_1 - V_i H y\| = \|V_i(\beta e_1 - H y)\| = \|\beta e_1 - H y\| \quad (2.29)$$

Dado que H en este caso tiene dimensión $i \times i$ y es no-singular porque A lo es, la norma minimizada vale exactamente 0 obtenida al resolver el sistema lineal $V_i H y = \beta e_1$ con respecto a y . Por lo tanto, cuando $h_{i+1,i} = 0$, GMRES encuentra la solución exacta del sistema $Ax = b$. Por otra parte, debe notarse que $i \leq n$ en V_i porque para generar un subespacio de dimensión n se necesitan como máximo n vectores linealmente independientes [IM98].

El método GMRES clásico (basado en el método de Arnoldi) para matrices no-simétricas es en general caro. GMRES converge a la solución exacta de (2.1) cuando calcula una base V_k con $k = n$ columnas.

2.3 GMRES con reinicio: GMRES(m)

A medida que k crece, la matriz de Arnoldi V_k se vuelve muy grande para ser almacenada y computada. Por ese motivo, como fue mencionado en la introducción (Capítulo 1), Saad y Schultz proponen reiniciar GMRES resultando en GMRES(m). [SS86]

En la j -ésima iteración, GMRES(m) usa el residuo r_{j-1} como residuo inicial, y ejecuta GMRES usando como máximo m vectores de una base ortogonal de $\mathcal{K}_m(A, r_{j-1})$, GMRES devuelve el vector z_j . Denotamos la aplicación de GMRES al vector r_{j-1} usando un subespacio de Krylov de dimensión m por: $z_{j-1} = GMRES(A, r_{j-1}, m)$. Entonces, la j -ésima solución x_j se calcula usando:

$$x_j = x_{j-1} + z_j. \quad (2.30)$$

GMRES(m) usa x_j para calcular un nuevo residuo inicial r_j y avanza a la siguiente iteración (i.e. la siguiente llamada a GMRES) hasta que un criterio de

parada sea satisfecho. Este procedimiento puede ser expresado en términos de las recurrencias:

$$\begin{cases} z_{j-1} &= \text{GMRES}(A, r_{j-1}, m), \\ x_j &= x_{j-1} + z_{j-1}, \\ r_j &= b - Ax_j. \end{cases} \quad (2.31)$$

La ventaja del procedimiento con reinicio es que se realizan como máximo m iteraciones del método de Arnoldi, de esta forma los costos de tiempo de ejecución y de memoria están bajo control.

Entrada:

A matriz del sistema,
 x_0 aproximación inicial,
 m dimensión del subespacio de Krylov en cada reinicio,
iteracionMaxima número máximo de reinicios

Salida: x resultado calculado

$x_0 = 0$

$r_0 = b - Ax_0$

// iteración de GMRES(m)

desde $j \leftarrow 1, \text{iteracionMaxima}$ **hacer**

$m = \text{obtenerSiguienteValorDeM}(k, r)$

$z_j = \text{GMRES}(A, r_{j-1}, m)$

$x_j = x_{j-1} + z_j$

$r_j = b - Ax_j$

salir Si el residuo cumple tolerancia indicada

fin_desde

retornar: x_j

Algoritmo 3: GMRES(m), Gmres con reinicio

Generalmente (en cada ciclo) el parámetro de reinicio m tiene un valor constante (quienes trabajan con GMRES(m) acostumbran usar $m = 30$). GMRES(m) corresponde al algoritmo (3).

En [SS86] Saad y Schultz presentaron el Teorema 4 que da una cota inferior para m . Sin embargo, como ellos mismos mencionan, m puede ser mucho menor que ese valor. También puede darse el caso en el que la cota inferior teórica sea mayor que n . En este caso, no es de utilidad porque sabemos que GMRES converge a la solución exacta cuando $m = n$.

Teorema 4 ([SS86] Cota inferior para m). *Suponer que hay v autovalores $\lambda_1, \lambda_2, \dots, \lambda_v$ de A con parte real no-positiva y el resto de los autovalores están encerrados en un círculo con centro en C con $C > 0$, y radio R , con $C > R$. Suponer también que A es diagonalizable $A = XDX^{-1}$. Sean $\kappa(X) = \|X\|\|X^{-1}\|$, $D = \max_{i=1, \dots, v; j=v+1, \dots, n} |\lambda_i - \lambda_j|$ y $d = \min_{i=1, \dots, v} |\lambda_i|$*

Si $m > v \log \left[\frac{DC}{dR} \kappa(X)^{1/v} \right] / \log \left[\frac{C}{R} \right]$ entonces, $GMRES(m)$ converge para cualquier vector inicial x_0

De acuerdo con [VL10], $GMRES(m)$ tiene convergencia sublineal para matrices normales. En [Zit08] son dadas condiciones generales para la convergencia de GMRES y $GMRES(m)$. Sin embargo, para matrices indefinidas $GMRES(m)$ no siempre converge [SS86]. También, como fue mencionado en el Capítulo 1, si el m apropiado no es elegido, la convergencia de $GMRES(m)$ no está garantizada [Emb03], y el método puede experimentar estancamiento ($r_{j+1} \approx r_j$ para todo j) o una convergencia lenta. Desafortunadamente, es difícil saber a priori qué valor de m utilizar. Si ocurre estancamiento, una estrategia simple puede ser aumentar el valor de m aunque no siempre funciona.

Capítulo 3

CONTROL Y GMRES(m)

En este capítulo incluimos algunas definiciones de la teoría de control. Seguidamente, introducimos el Controlador PID (Proporcional-Integral-Derivativo). Luego, hablamos de los algoritmos iterativos utilizando el método de la Función de Control de Lyapunov (FCL) [BK04] y lo aplicamos al algoritmo iterativo conocido como Orthomin. El uso de una FCL ha demostrado ser efectivo para algoritmos con recurrencias cortas. Sin embargo, GMRES está entre los algoritmos que utilizan recurrencias largas [Saa03, CB04]. GMRES primero construye una base ortogonal de vectores y luego minimiza la norma euclidiana (l_2) del residuo expresado usando esa base. Finalizado este proceso puede calcular x_{k+1} .

El objetivo es controlar el parámetro de reinicio m , por ese motivo presentamos controladores discretos PID (proporcional, integrativo derivativo). La estrategia propuesta utiliza una recurrencia inspirada en un controlador PD (proporcional, derivativo) y es presentada en la sección 3.5.3, el pseudocódigo correspondiente está en la sección 4.2.1. Inicialmente GMRES(m) es modelado de una forma similar a un algoritmo iterativo genérico.

Además, en este contexto mostramos que la estrategia de Baker [BJK09] puede ser vista como un controlador Derivativo donde la derivada es constante.

3.1 Algunas definiciones

Las definiciones fueron obtenidas de [Oga87].

Variable controlada: La cantidad o condición que es medida y controlada. Normalmente es la salida del sistema.

Variable manipulada: La cantidad o condición que es modificada para afectar el valor de la variable controlada.

Controlar: Significa medir el valor de la variable controlada y aplicar la varia-

ble manipulada para corregir la desviación de la variable controlada con respecto al valor deseado.

Planta: Es una pieza de equipo, un conjunto de partes de máquina cuyo propósito es realizar una tarea particular. Cualquier objeto físico a ser controlado

Sistema: Es un conjunto de partes o componentes que funcionan juntas para realizar un objetivo particular. Un componente tiene una función particular en el sistema del que forma parte.

Perturbación: Es una señal que afecta desfavorablemente la señal de salida del sistema.

Control realimentado: Se refiere a la operación en la que en presencia de perturbaciones tiende a reducir la diferencia entre la salida del sistema y el valor de referencia, usando como base esa diferencia.

Sistema de Control realimentado: Es un sistema que mantiene una relación prescrita entre la salida y alguna entrada de referencia comparándolas y usando la diferencia como medio de control.

Sistemas de Control de Lazo Cerrado: En la práctica, sistema de control realimentado y sistema de control de bucle cerrado son términos intercambiables. En un sistema de control de lazo cerrado el error, que es la diferencia entre la señal de entrada y la señal realimentada (que es la señal de salida), es realimentado al controlador para reducir el error y llevar la salida del sistema al valor deseado.

Sistemas de Control de Lazo Abierto: En estos sistemas, la salida no afecta a la acción de control. Cualquier sistema de control que opere en regímenes de tiempo es un sistema de bucle abierto. Su funcionamiento depende de una buena calibración y en general no responde bien a perturbaciones.

Diagrama de Bloques: Es una representación gráfica de las funciones realizadas por cada componente y el flujo de señales. En un diagrama de bloques las variables están concatenadas unas con otras por bloques funcionales. Un bloque funcional o simplemente un bloque representa la operación que realiza el bloque sobre la entrada para producir la salida. Las funciones de transferencia se ponen en bloques y las señales pasan solamente en el sentido de las flechas. Entre las ventajas del uso de un diagrama de bloques se tienen:

- permite visualizar la operación total del sistema más rápidamente que usando un modelo de ecuaciones del sistema.
- facilita tener una imagen del sistema completo donde se puede evaluar la contribución de cada componente.

3.2 Control PID

Esta sección contiene información obtenida de [Vis06].

Consideramos un sistema de control realimentado con un controlador C y una planta P. El sistema tiene una señal de referencia $r(t)$ y una señal de salida $y(t)$, $u(t)$ es la señal de control, $d(t)$ representa perturbaciones y $n(t)$ errores de medición de la salida $y(t)$.

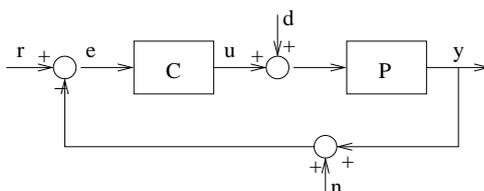


Figura 3.1: Sistema de Control Realimentado.

Un sistema de control PID aplica de forma adecuada tres tipos de acciones de control: una acción proporcional, una acción integrativa y una acción derivativa.

La señal de control $u(t)$ correspondiente a la acción proporcional es proporcional al error actual $e(t) = r(t) - y(t)$. Una ley de control proporcional pura es:

$$u(t) = K_p e(t) = K_p (r(t) - y(t))$$

K_p se conoce como la ganancia proporcional. La operación de la acción proporcional es aumentar la variable de control cuando el error actual es grande.

Por otra parte, la acción integral considera los errores anteriores. Una ley de control integral pura es:

$$u(t) = K_i \int_0^t e(\tau) d\tau$$

K_i se conoce como la ganancia integral. Un controlador integral puro reduce el error de estado estable a cero.

Finalmente, la acción derivativa está basada en la predicción de los valores futuros del error. Una ley de control derivativo pura es:

$$u(t) = K_d \frac{de(t)}{dt}$$

K_d se conoce como la ganancia derivativa. Para entender mejor la acción derivativa, consideramos la aproximación del error $e(t+h)$ usando dos términos de una serie de Taylor.

$$e(t+h) = e(t) + h \frac{de}{dt}$$

Si consideramos la expresión anterior como parte de una ley de control proporcional tenemos:

$$u(t) = K_p(e(t) + h \frac{de}{dt})$$

Esto resulta en un controlador PD. En este caso, la variable de control en el tiempo t está basada en el aproximado para el tiempo $t + h$. Por ese motivo, la acción derivativa es conocido como control anticipativo, tasa de acción o pre-acción. La acción derivativa puede anticipar una tendencia incorrecta del error $e(t)$ y corregirla.

Un controlador PID en forma paralela utiliza a la vez las acciones proporcional, integral y derivativa con constantes K_p , K_i y K_d independientes entre sí.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \int_0^t \frac{de(t)}{dt} \quad (3.1)$$

Las tres acciones tienen sus problemas, que pueden ser resueltos con una elección adecuada de las constantes:

- La parte proporcional puede oscilar cuando llega al estado estable aún cuando haya un polo en el origen.
- La parte integral puede solucionar el problema de oscilación de la parte proporcional. Sin embargo, la parte integral tiene problemas cuando el error disminuye lentamente durante la respuesta transitoria. Aún cuando la variable del proceso llega al nivel de referencia deseado, el controlador está saturado por el término integral y esto conduce a un sobrepaso que puede ser prolongado.
- La parte derivativa actúa en las frecuencias altas amplificando el error de las mediciones $n(t)$ (ver Figura 3.1). Otro problema con la acción derivativa es que cuando hay cambios abruptos en los valores de $e(t)$, la derivada tiene un pico que hace que la variable de control tome valores grandes. En general, esto no es deseable.

Para una información más detallada, consultar [Vis06].

3.3 Control y Algoritmos Iterativos

De manera a formular la resolución iterativa de un sistema lineal, consideramos el residuo $r_j = b - Ax_j$ como el error o desviación [BK04]. Note que llevar a cero el residuo r_j corresponde a resolver el sistema lineal $Ax = b$. El residuo es

llevado a cero y la solución x_j es llevada a la solución del sistema por medio de una secuencia apropiada de vectores x_j .

Dado un sistema lineal de ecuaciones, en general, un método iterativo puede ser descrito como una sucesión de aproximaciones x_j de la forma:

$$\begin{cases} z_{j-1} = \text{Solver}, \\ x_j = x_{j-1} + z_{j-1}, \\ r_j = b - Ax_j. \end{cases} \quad (3.2)$$

donde el *Solver* es una función que devuelve la actualización a partir de un determinado residuo. El *Solver* puede ser interpretado como un operador M^{-1} que aproxima la inversa de la matriz A

$$\begin{aligned} z_{j-1} &= \text{Solver}(r_{j-1}), \\ z_{j-1} &= M^{-1}r_{j-1}, \\ z_{j-1} &= M^{-1}(b - Ax_{j-1}). \end{aligned} \quad (3.3)$$

Note que el método iterativo (3.2) usando (3.3) toma la forma:

$$\begin{aligned} x_j &= x_{j-1} + M^{-1}(b - Ax_{j-1}), \\ x_j &= (I - M^{-1}A)x_{j-1} + M^{-1}b. \end{aligned}$$

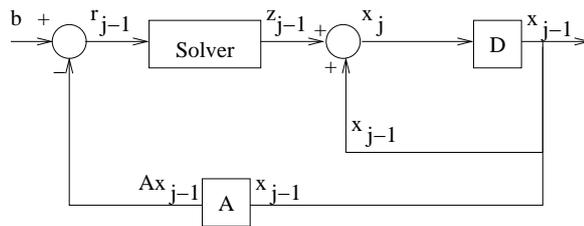


Figura 3.2: Diagrama de bloques para un algoritmo iterativo.

La Figura 3.2 muestra un diagrama de bloques de un algoritmo iterativo genérico. Las flechas indican el flujo de los datos.

En términos de control para conducir el residuo a cero se debe encontrar una secuencia de control z_j que lleve la salida (esto es el residuo r_j) a cero, y como consecuencia lleve la variable de estado x_j a la solución deseada.

Para los métodos iterativos de resolución de sistemas de ecuaciones lineales que utilizan recurrencias cortas, el método de control que usa una función de

Control de Lyapunov (FCL) ha demostrado ser un mecanismo unificador y capaz de ofrecer deducciones de muchos de ellos. Orthomin y otros métodos de recurrencias cortas como Gradiente Conjugado, han sido deducidos utilizando sistemas bilineales y funciones de control de Lyapunov [BK04]. Esto constituye un avance importante para el uso de la teoría de Control en el contexto de la resolución de $Ax = b$.

Incluimos un ejemplo de recurrencias cortas para el método conocido como Orthomin. Las recurrencias para este método son

$$r_{k+1} = r_k - \alpha_k Ar_k x_{k+1} = (I - \alpha_k A)x_k + \alpha_k b \quad (3.4)$$

El sistema dinámico $r_{k+1} = r_k - \alpha_k Ar_k$ es bilineal: Si α_k es constante, es lineal para r_k . Si r_k es constante, es lineal para α_k . Usando como función Lyapunov

$$V(r_k) = \langle r_k, r_k \rangle = r_k^T r_k \quad (3.5)$$

y calculando $\delta V = V(r_{k+1}) - V(r_k)$ de tal forma que sea negativa definida obtenemos:

$$\alpha_k = \frac{\langle r_k, Ar_k \rangle}{\langle Ar_k, Ar_k \rangle} \quad (3.6)$$

El uso de una función de control de Lyapunov negativa definida garantiza que r_k converge monotónicamente a cero cuando $k \rightarrow \infty$. Es decir, el método converge a la solución del sistema lineal. Para que $V(r_k)$ sea negativa definida, se debe evitar que $\langle r_k, Ar_k \rangle = 0$. Esto es equivalente a decir que cero no pertenece a $\mathcal{F}(A) = \{y^H A y : y \in \mathbb{C}^n, y^H y = 1\}$ (campo de valores de A definido en [Gre97]). En términos prácticos, si $\langle r_k, Ar_k \rangle \approx 0$, el algoritmo Orthomin presenta una convergencia lenta o un estancamiento de la solución.

3.4 LQRES

Debemos mencionar que utilizando otras herramientas de la teoría de Control, Helmke y Jordan [HJL06] han presentado un método denominado LQRES que resuelve una ecuación algebraica de Riccati. Este método es estable, pero los mismos autores indican que para matrices con dimensión mayor que $n = 1000$ resolver la ecuación de Riccati se vuelve intratable. Por otra parte, aún cuando la solución de la ecuación de Riccati sea aproximada, el parámetro B es matricial y encontrarlo no es trivial. Para una discusión detallada de este algoritmo ir al Apéndice B (§B).

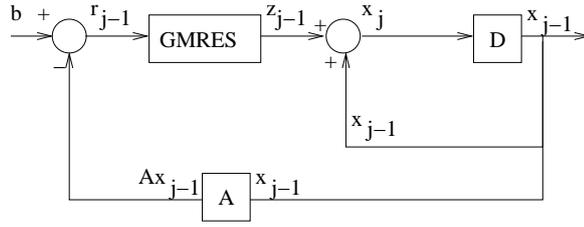


Figura 3.3: Diagrama de bloques para $GMRES(m)$.

3.5 $GMRES(m)$ con m controlado

GMRES no tiene una recurrencia directa o corta, primero realiza una ortogonalización y luego resuelve un problema de mínimos cuadrados. Por este motivo, no es trivial utilizar el método bilineal con función de control de Lyapunov para estabilizar GMRES.

A continuación modelamos $GMRES(m)$ como un sistema de control realimentado para luego acoplar a ese sistema un controlador para el número de reinicios m . Nuestro objetivo es cambiar el valor del reinicio en cada llamada a GMRES, el valor que vamos a usar en cada llamada estará dado por un controlador inspirado en un controlador PD, (proporcional derivativo).

En el contexto de $GMRES(m)$, la relación entre el residuo r_{j-1} y z_{j-1} está dada por la subrutina GMRES con el parámetro interno fijo m . Las recurrencias asociadas a este modelo realimentado son

$$\begin{cases} z_{j-1} &= GMRES(A, r_{j-1}, m), \\ x_j &= x_{j-1} + z_{j-1}, \\ r_j &= b - Ax_j. \end{cases} \quad (3.7)$$

Un diagrama de bloques asociado a las recurrencias (3.7) que modelan $GMRES(m)$ es presentada en la Figura 3.3.

La expresión de z_{j-1} es una función del residuo r_{j-1} y del parámetro de reinicio m . Para hacer notar que no se necesita usar un valor fijo de m , denotamos el parámetro de reinicio por m_{j-1} . Entonces, podemos escribir $z_{j-1} = GMRES(A, r_{j-1}, m_{j-1})$. Para la llamada a GMRES, r_{j-1} es el residuo inicial y m_{j-1} indica la dimensión máxima del subespacio de Krylov. Si m_{j-1} toma un valor mayor, suficiente para que $GMRES(m)$ realice una única iteración, se tiene GMRES. Hacemos notar esto para remarcar la estructura iterativa de $GMRES(m)$.

Para mejorar z_{j-1} , dado que la subrutina GMRES tiene los parámetros residuo r_{j-1} (residuo) y m_{j-1} (número de ortogonalizaciones), podríamos tratar de

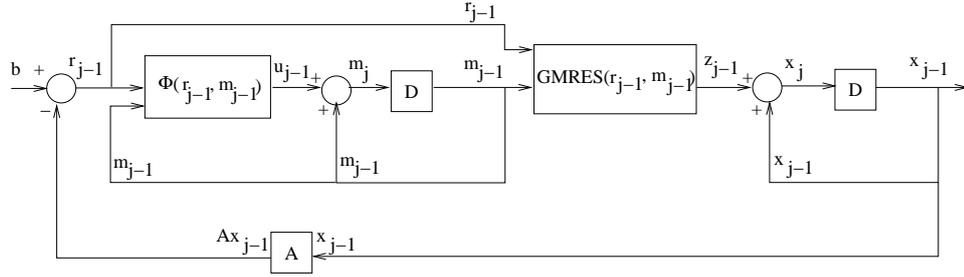


Figura 3.4: Diagrama de bloques para GMRES(m) adaptativo.

modificar uno o ambos. Varias técnicas han sido presentadas en la literatura, por ejemplo [GRLB08] cambia solamente r_{j-1} usando un residuo híbrido y con eso consiguen evitar el estancamiento en problemas específicos. [BJK09, GC05, SWK98, ZN05, MN00, HN03] cambian solamente m_{j-1} . Note que m está fuertemente relacionada al costo computacional del método. En esta tesis modificamos el parámetro m_{j-1} de forma adecuada y adaptativa para mejorar el valor de z_{j-1} .

Para actualizar m_{j-1} , usamos una recurrencia similar a la usada para x_j en (3.7)

$$m_j = m_{j-1} + u_{j-1} \quad (3.8)$$

donde u_{j-1} es el parámetro de adaptación o la *variable de control*.

Una ley general de control retroalimentado para u_{j-1} tiene la forma:

$$u_{j-1} = \phi(r_q, m_{j-1}). \quad (3.9)$$

donde usamos índice q en lugar de $j-1$ para indicar que la función ϕ puede relacionar el residuo r_{j-1} y los residuos previos ($r_{j-2}, r_{j-3}, \dots, r_{j-k}$). Un diagrama de bloques es presentado en la Figura (3.4).

Para controlar m introducimos en la siguiente subsección la medida de convergencia que utilizamos. Luego presentamos nuestra propuesta inspirada en un controlador PD. Finalmente mostramos que la estrategia de Baker [BJK09], puede ser vista como un controlador D (derivativo) donde la derivada es constante.

3.5.1 Medida de Convergencia

El ángulo entre residuos consecutivos r_{j-1} y r_{j-2} de GMRES(m) es denotado por $\angle(r_{j-1}, r_{j-2})$ y es conocido como ángulo secuencial.

El ángulo secuencial está relacionado con la disminución en la norma del

residuo por la expresión:

$$\cos(\angle(r_{j-1}, r_{j-2})) = \frac{\|r_{j-1}\|}{\|r_{j-2}\|}. \quad (3.10)$$

Para una demostración de que la expresión anterior corresponde efectivamente a $\cos(\angle(r_{j-1}, r_{j-2}))$ ver [BJM05] o (1) en el apéndice. Cuando GMRES(m) tiene una buena convergencia, r_{j-1} y r_{j-2} son casi perpendiculares y $\cos(\angle(r_{j-1}, r_{j-2})) \approx 0$. En una situación de convergencia pobre, r_{j-1} y r_{j-2} son casi paralelos o linealmente dependientes, y $\cos(\angle(r_{j-1}, r_{j-2})) \approx 1$. Es decir, según el valor de $\cos(\angle(r_{j-1}, r_{j-2}))$ podemos identificar si GMRES tiene una convergencia aceptable o no.

Describimos a continuación algunas funciones $\phi(\cdot, \cdot)$ para la ley (3.9).

3.5.2 Justificación de la variación del parámetro de reinicio

En todos los trabajos la justificación de la variación del valor de m es fundamentalmente empírica. En [BJK09] observaron que cuando hay estancamiento o convergencia pobre, los residuos intercalados de GMRES(m) a menudo apuntan casi en la misma dirección. Es decir, $r_{j+1} \approx \sigma r_{j-1}$ para algún $\sigma \leq 1$. Las soluciones difieren muy poco cuando los residuos son paralelos. Este comportamiento retrasa el tiempo de convergencia. En [BJK09] demuestran que este comportamiento se cumple para matrices simétricas. Sin embargo, también lo observaron en matrices generales no-hermitianas.

3.5.3 Controlador Proporcional Derivativo PD(m)

Los ingenieros de Control hace tiempo usan un método conocido como controlador Proporcional-Integrativo-Derivativo (PID) para procesos.[Vis06]. Nosotros consideramos un controlador PID discreto con la forma:

$$u_{j-1} = \alpha \|r_{j-1}\| + \beta \sum_{i=1}^{j-1} \|r_i\| + \gamma \left(\frac{\|r_{j-1}\| - \|r_{j-3}\|}{2} \right), \quad (3.11)$$

donde α , β y γ son constantes reales denominadas constantes proporcional, integrativa y derivativa respectivamente. Note que en (3.11) utilizamos un esquema de diferencias finitas para la aproximación de la derivada.

Para trabajar con variables independientes de la dimensión, y con la intención de definir el controlador PID en términos de ángulos secuenciales consecutivos

$\angle(r_{j-1}, r_{j-2})$ y $\angle(r_{j-2}, r_{j-3})$ dividimos el controlador u_{j-1} en el tiempo t_{j-1} por la norma del residuo $\|r_{j-2}\|$. Entonces, considerando (3.11), la expresión (3.8) toma la forma:

$$m_j = m_{j-1} + \alpha \frac{\|r_{j-1}\|}{\|r_{j-2}\|} + \beta \sum_{q=1}^{j-1} \frac{\|r_q\|}{\|r_{j-2}\|} + \gamma \left(\frac{\|r_{j-1}\| - \|r_{j-3}\|}{2\|r_{j-2}\|} \right). \quad (3.12)$$

Además, dado que m es un entero, tomamos la parte entera de los términos proporcional, integrativo y derivativo.

$$m_j = m_{j-1} + \left\lfloor \alpha \frac{\|r_{j-1}\|}{\|r_{j-2}\|} + \beta \sum_{q=1}^{j-1} \frac{\|r_q\|}{\|r_{j-2}\|} + \gamma \frac{\|r_{j-1}\| - \|r_{j-3}\|}{2\|r_{j-2}\|} \right\rfloor. \quad (3.13)$$

En este trabajo consideramos una forma más simple de (3.13) conocida como controlador proporcional derivativo, aquí lo denominamos PD(m). La formulación PD(m) se obtiene haciendo $\beta = 0$. Restringimos α y γ a valores enteros. in (3.13):

$$m_j = m_{j-1} + \left\lfloor \alpha \frac{\|r_{j-1}\|}{\|r_{j-2}\|} + \gamma \frac{\|r_{j-1}\| - \|r_{j-3}\|}{2\|r_{j-2}\|} \right\rfloor \quad (3.14)$$

Tomamos la estrategia de disminuir m_j . Al variar m_j esperamos evitar residuos intercalados paralelos $r_{j+1} \approx \sigma r_{j-1}$ que enlentecen la convergencia. También mencionamos que en el controlador PD(m), el término proporcional actúa principalmente cuando el término derivativo está inactivo y viceversa. Con relación a la parte derivativa, es importante notar que cuando hay buena convergencia $(\|r_{j-1}\| - \|r_{j-3}\|)/\|r_{j-2}\|$ se vuelve importante y esta parte modifica el valor de m_j . Cuando la convergencia es baja, $(\|r_{j-1}\| - \|r_{j-3}\|)/\|r_{j-2}\| \approx 0$ y su influencia en m_j es mínima. Cambiar m cuando hay buena convergencia no es malo, no cambiar m cuando hay buena convergencia, no implica buena convergencia en la siguiente iteración. Para la parte proporcional, ver la discusión sobre la ecuación (3.10).

En conclusión, el término proporcional modifica m_j cuando GMRES(m) no converge y el término derivativo cambia m_j cuando GMRES(m) converge. Note que una buena convergencia en una iteración no garantiza una buena convergencia en la siguiente iteración. Puede pensarse que el algoritmo ya convergió lo que podía en la dirección dada por la iteración y a partir de allí solo realice mejoras pequeñas. Es decir, en la estrategia propuesta, m_j es modificado en cada iteración.

Observación Dado que $0 \leq \cos(\angle(r_{j-1}, r_{j-2})) \leq 1$, el controlador proporcional

(cuando $\gamma = 0$) se asemeja a la regla propuesta por Baker et al. [BJK09] (ver el algoritmo 5), solo que no necesita preguntar por la tasa de convergencia, su efecto es alto solo cuando la tasa de convergencia es mala.

Ajuste del controlador PD(m). Aquí discutimos la heurística para seleccionar α y γ . Buscamos experimentalmente valores de α y γ (§5.2).

- Los valores probados fueron: $\alpha \in [2, 9]$ y $\gamma \in [-9, -2]$. Los valores para γ son negativos porque buscamos disminuir m y los valores de las normas de los residuos generados por GMRES(m) forman una sucesión de valores monotónicamente no crecientes.
- Ejecutamos PD-GMRES con 64 pares (α, γ) para cada problema y medimos los tiempos de ejecución. También ejecutamos GMRES(30) para cada problema. Usamos el tiempo relativo que es el cociente entre $\frac{t_{PD-GMRES(m)}}{t_{GMRES(30)}}$. Buscamos el par de enteros (α, γ) que minimice la sumatoria de tiempos relativos de los problemas.

$$\min_{(\alpha \in [2,9], \gamma \in [-9,-2])} \sum_{prob=1}^{15} \frac{t(PD - GMRES(m))}{t(GMRES(30))} \quad (3.15)$$

- Seleccionamos el par $(3, -5)$ a pesar de que el mejor según el criterio anterior fue el par $(3, -9)$. Las diferencias entre los tiempos de ejecución usando ambos pares no son significativas. Además, con un valor de -5 para γ damos un peso menor a la derivada. La derivada de las normas residuales es importante cuando hay buena convergencia. En este caso el valor -5 hace que el cambio de m sea más conservador que usando el valor -9 . (§5.3)

Usando las consideraciones previas, la recurrencia que utilizamos para m es:

$$m_j = m_{j-1} + \left\lfloor \left(-3 \frac{\|r_{j-1}\|}{\|r_{j-2}\|} + 5 \frac{\|r_{j-1}\| - \|r_{j-3}\|}{2\|r_{j-2}\|} \right) \right\rfloor \quad (3.16)$$

El controlador PD(m) propuesto comienza con un valor inicial $m = m_{initial}$ y tiende a disminuir m . Sin embargo, si $m \leq m_{min}$, el valor de m para la siguiente iteración de GMRES(m) es $m = m_{initial} + m_{step}$ y también se cambia el valor inicial $m_{initial} = m_{initial} + m_{step}$. No imponemos un máximo m_{max} . El pseudocódigo para la regla propuesta está en el Algoritmo (4). Usamos los parámetros: $m_{initial} = 30$, $m_{step} = 3$ and $m_{min} = 1$.

3.5.4 Controlador Constante

En [BJK09] m puede fluctuar entre m_{\min} y m_{\max} . En la iteración j el reinicio m_j es tal que $m_{\min} \leq m_j \leq m_{\max}$. Si la convergencia no es buena, la regla decrementa m en una constante $u_{j-1} = d$. El código de esta regla está en el algoritmo (5).

A continuación describimos la regla utilizada en [BJK09]. El valor inicial para m es el máximo m_{\max} . La medida de convergencia es el valor de $\cos(\angle(r_{j-1}, r_{j-2}))$. Si el ángulo secuencial es grande (cercano a 90°), indica buena convergencia, $m_j = m_{j-1}$. Si el ángulo secuencial es pequeño (cercano a 0°), indica baja convergencia y el parámetro de reinicio es disminuido $m_j = m_{j-1} - d$. Si $m_j \leq m_{\min}$, m_j se pone en m_{\max} . Entonces, cuando no hay convergencia, esta estrategia utiliza la ley:

$$m_j = m_{j-1} - d. \quad (3.17)$$

3.6 GMRES(m) con m aleatorio

Antes de pasar a las pruebas de los métodos, debemos mencionar que usando un valor aleatorio para m entre los valores m_{\min} y m_{\max} obtenemos

$$m_j = \text{random}(m_{\min}, m_{\max}) \quad (3.18)$$

La regla (3.18) calcula la solución en menos tiempo que el uso de un valor fijo para m . Sin embargo, tiene un tiempo mayor que la regla de Baker.

Incluimos un ejemplo de esta regla comparada con GMRES(m) para el problema Wang2 (problem 13 from table (4.1)). Note que en este ejemplo, aunque la alternativa aleatoria realiza más iteraciones, halla la solución en un 60% del tiempo usado por GMRES(m).

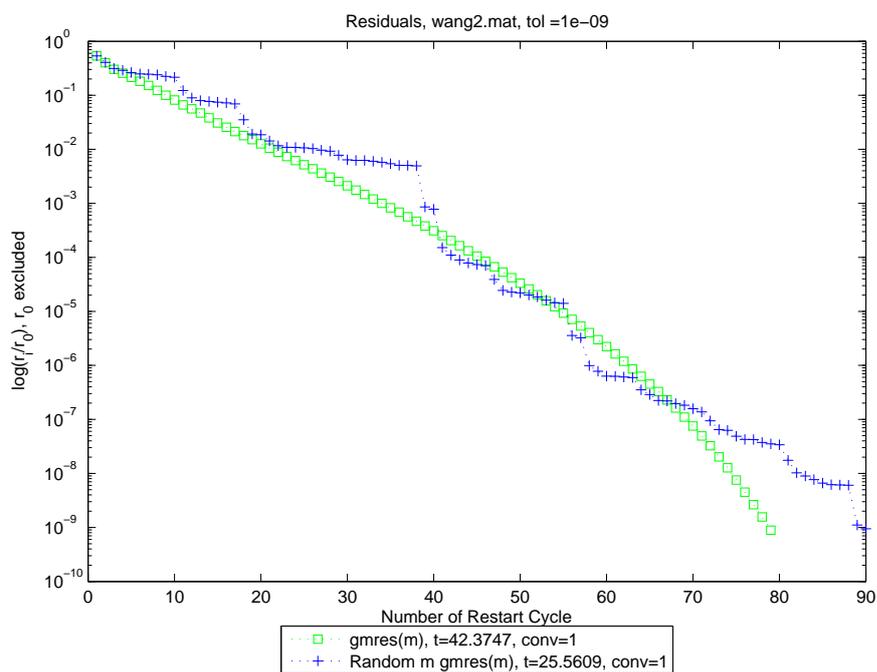


Figura 3.5: Matriz Wang2: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

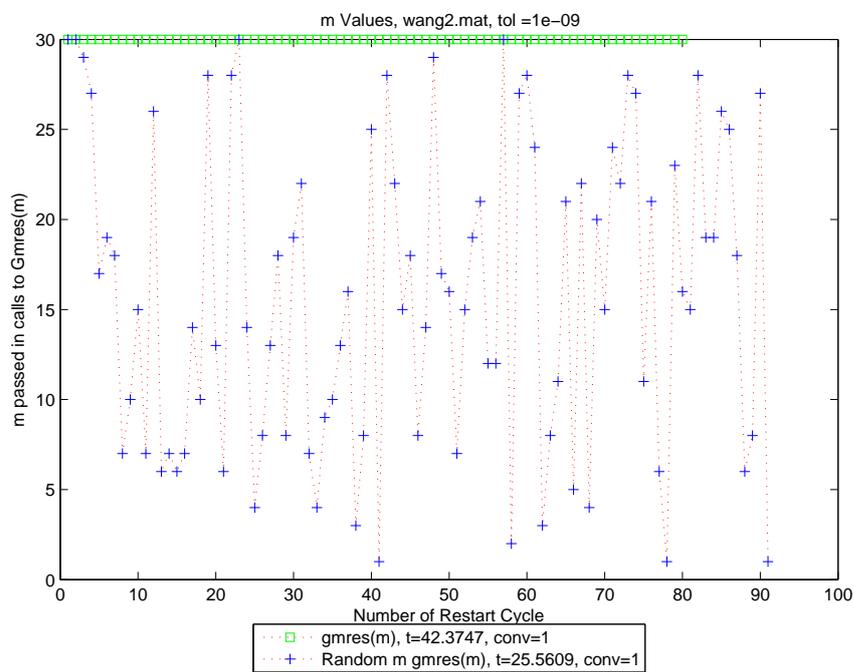


Figura 3.6: Matriz Wang2: Reinicios aleatorios m Vs Iteraciones.

Capítulo 4

EXPERIMENTOS NUMÉRICOS

4.1 Resumen de los resultados

Los sistemas lineales $Ax = b$ resueltos utilizan matrices A obtenidas del repositorio de la Universidad de Florida [bTDFUoFfAR], que incluye matrices del Mercado de Matrices [MfNU]. Las matrices seleccionadas son matrices no-simétricas y corresponden a una variedad de problemas reales. Para los problemas en los que falta el lado derecho b , el mismo fue generado aleatoriamente usando una distribución uniforme con valores entre los valores mínimo y máximo $(A)_{i,j}$ de A . Los problemas usados en los experimentos están listados en la Tabla 4.1.

La resolución de los sistemas de ecuaciones fue realizada usando el programa Matlab para Linux en una computadora de escritorio con un procesador Intel Core 2 Duo y 2Gb de memoria RAM.

Tabla 4.1: Lista de problemas. n es la dimensión de la matriz y nnz es el número de coeficientes distintos de cero

	Problema	n	nnz	Area de Aplicación
1	add20	2395	17319	Diseño de Componente de Computador
2	cdde1	961	4681	Operador 2D de convección-difusión
3	circuit_2	4510	21199	Simulación de circuitos
4	fpga_trans_01	1220	7382	Simulación de circuitos
5	orsirr_1	1030	6858	Simulación de reservorio de Petróleo
6	orsreg_1	2205	14133	Simulación de reservorio de Petróleo
7	pde2961	2961	14585	Modelo para Ec. Dif. Parcial
8	raefsky1	3242	294276	Flujo de Fluido Incompresible
9	raefsky2	3242	293551	Flujo de Fluido Incompresible
10	rdb2048	2048	12032	Modelo Brusselator para Reacción-difusión
11	sherman4	1104	3786	Simulación de reservorio de Petróleo
12	steam2	600	13760	Vapor inyectado para extraer petróleo
13	wang2	2903	19093	Ecuaciones de continuidad de electrones
14	watt_1	1856	11360	Ingeniería de Petróleo
15	young3c	841	3988	Dispersión acústica

Resultados Comparativos

De manera a comparar los resultados obtenidos al resolver el sistema $Ax = b$ con la estrategia propuesta en esta tesis, implementamos los siguientes algoritmos:

1. GMRES(m) clásico con $m = 30$.
2. la regla de Baker (obtenida de [BJK09]).
3. la regla de González (obtenida de [GC05]).
4. la regla inspirada en control, propuesta en esta tesis, denotada por PD-GMRES.

La configuración usada para los algoritmos es:

- La solución inicial es $x_0 = 0$, que es equivalente a $r_0 = b$.
- El criterio de parada en la iteración j –ésima es $\frac{\|r_j\|}{\|r_0\|} < 10^{-9}$.
- Para PD-GMRES (ver algoritmo (4)) usamos los parámetros, $m_{initial} = 30$, $m_{step} = 3$, $m_{min} = 1$.
- Para Alg. 1 (la regla de Baker, ver algoritmo (5)) usamos los parámetros $m_{min} = 1$, $m_{max} = 30$ y $m_{step} = 3$.

- Para Alg. 2 (la regla de Gonçalez, ver algoritmo (6)) usamos los parámetros $m_{ini} = 10$, $m_{max} = 30$ and $tolerancia = 10^{-9}$.

La figura 4.1 resume los resultados experimentales. Para evaluar los algoritmos, los ejecutamos cinco veces y promediamos sus tiempos de ejecución. Para evitar diferencias de escala entre los problemas, los tiempos de ejecución mostrados son relativos al tiempo de ejecución de GMRES(30). Por eso, 1 corresponde a GMRES(30), los valores por debajo de 1 corresponde a tiempos de ejecución menores que el tiempo de GMRES(30) y los valores por encima de 1 son peores que el tiempo de GMRES(30). Las barras están agrupados por problema. El número de problema corresponde a los números listados en la Tabla 4.1. Notar que todos los algoritmos convergen para todos los problemas, excepto para el problema 15. Para este problema, solo PD-GMRES converge. Además, PD-GMRES tiene mejor tiempo de ejecución que los algoritmos probados excepto para los problemas 7 y 10, donde tiene un tiempo similar al de GMRES(30).

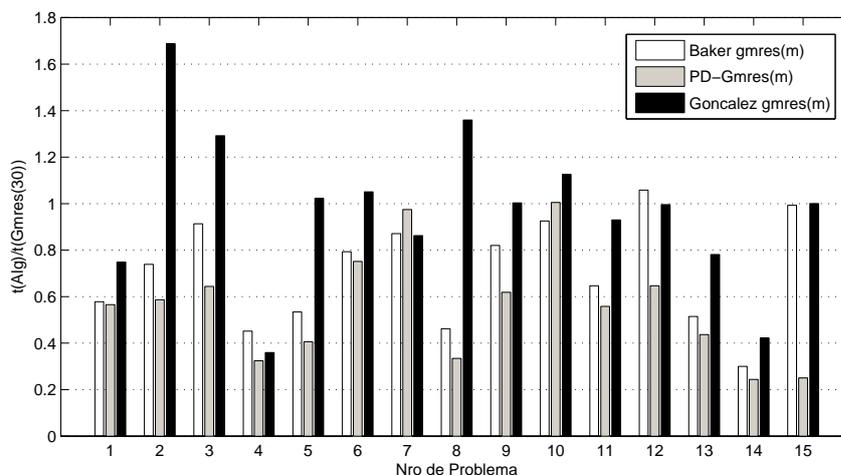


Figura 4.1: Tiempos de ejecución relativos. Las barras están agrupadas por problemas. Cuanto más baja la altura de la barra, es mejor el tiempo que lleva hasta encontrar la solución. En el problema 15, solo PD-GMRES convergió. Alg. 1 corresponde a la regla de Baker y Alg. 2 a la regla de Gonçalez.

Como complemento a la figura de barras, incluimos la siguiente tabla con los tiempos en segundos. Los tiempos reportados corresponden al promedio de 5 ejecuciones de los métodos iterativos evaluados.

Nro de Problema	GMRES(30)	Baker gmres(m)	PD-Gmres(m)	Goncalez gmres(m)
1	12.6558	7.3204	7.1589	9.4774
2	2.0330	1.5027	1.1928	3.4308
3	33.7340	30.8152	21.7160	43.5941
4	11.4255	5.1627	3.7008	4.1089
5	13.9726	7.4597	5.6753	14.2852
6	7.6170	6.0429	5.7252	8.0030
7	7.1229	6.2062	6.9418	6.1424
8	75.7676	35.0004	25.3235	103.0190
9	104.7672	85.9391	64.8294	105.1181
10	4.9007	4.5343	4.9243	5.5155
11	2.2970	1.4858	1.2823	2.1369
12	0.3655	0.3867	0.2363	0.3640
13	42.2690	21.7541	18.4714	33.0199
14	29.3296	8.7968	7.1647	12.3781
15	57.2233	56.8289	14.3143	57.2454

4.2 Reglas implementadas

4.2.1 PDRULE

Regla propuesta en esta tesis. Los parámetros utilizados fueron: $m_{initial} = 30$, $m_{step} = 3, m_{min} = 1$.

Entrada:

m_{j-1} último valor de m ,

$m_{inicial}$ último valor inicial de m ,

m_{minimo} valor mínimo de m ,

r vector de residuos,

j el valor retornado será el reinicio para la iteración j ,

m_{paso} valor con el que se incrementa el valor inicial de m cuando se llega a un valor menor o igual que m_{minimo}

Salida: m_j parámetro de reinicio para la iteración j

si ($j > 3$) **entonces**

$$m_j = m_{j-1} + \left[\left(-3 \frac{\|r_{j-1}\|}{\|r_{j-2}\|} + 5 \frac{\|r_{j-1}\| - \|r_{j-3}\|}{2\|r_{j-2}\|} \right) \right]$$

si_no si ($j > 2$) **entonces**

$$m_j = m_{j-1} + \left[-3 \frac{\|r_{j-1}\|}{\|r_{j-2}\|} \right]$$

si_no

$$m_j = m_{inicial}$$

fin_si

si ($m_j < m_{minimo}$) **entonces**

$$m_{inicial} = m_{inicial} + m_{paso}$$

$$m_j = m_{inicial}$$

fin_si

return m_j

Algoritmo 4: Regla PD. Calcula el parámetro de reinicio usando una fórmula inspirada en un controlador PD.

4.2.2 Regla de Baker

Regla propuesta en [BJK09].

Los parámetros utilizados fueron: $m_{\text{minimo}} = 1$, $m_{\text{maximo}} = 30$ and $\text{paso} = 3$.

Entrada:

m_{j-1} último valor de m ,

m_{inicial} último valor inicial de m ,

m_{minimo} valor mínimo de m ,

m_{maximo} valor máximo de m ,

r residuos anteriores,

j iteración para la cual se calcula el valor de reinicio,

paso valor en el que se disminuye m_{j-1} si la convergencia no es aceptable

Salida: m_j parámetro de reinicio para la iteración j

$$\text{tasa} = \frac{\|r_{j-1}\|}{\|r_{j-2}\|}$$

$$\text{tasaMaxima} = \cos(8/90 * \pi/2) \quad // \quad \cos(8)$$

$$\text{tasaMinima} = \cos(8/9 * \pi/2) \quad // \quad \cos(80)$$

si ($\text{tasa} > \text{tasaMaxima}$) **entonces**

$$m_j = m_{\text{maximo}}$$

si_no si ($\text{tasa} < \text{tasaMinima}$) **entonces**

$$m_j = m_{j-1}$$

si_no

adjust

si ($(m_{j-1} - \text{paso}) \geq m_{\text{minimo}}$) **entonces**

$$m_j = m_{j-1} - \text{paso}$$

si_no

$$m_j = m_{\text{maximo}}$$

fin_si

fin_si

return m_j

Algoritmo 5: Regla de Baker. Si la convergencia no es aceptable, disminuye el valor de m .

4.2.3 Regla de Gonçalez

Regla propuesta en [GC05].

Los parámetros utilizados fueron: $m_{ini=10}$, $m_{max} = 30$, $tolerance = 10^{-9}$.

Entrada:

m_{j-1} último valor de m ,

$m_{inicial}$ valor inicial de m ,

m_{maxima} valor máximo de m ,

r residuos anteriores,

j iteración para la cual se calcula el reinicio,

$tolerancia$ tolerancia utilizada como criterio de parada de GMRES(m)

Salida: m_j valor de reinicio para la iteración j

$m_{inicial} = 10$

si ($mod((j - 1), 5) == 0$) **entonces**

si ($\log_{10}(\|r_{j-1}\|) > 0$) **entonces**

$m_j = \min(2 * m_{j-1}, m_{maxima})$

si_no

si ($\log_{10}(\|r_{j-1}\|) \leq 0$ and $\log_{10}(\|r_{j-1}\|) > (2 \log_{10}(tolerancia)/3)$)

entonces

si ($(j - 1) > 5$ and $\frac{\|r_{(j-1)-5}\|}{\|r_{j-1}\|} > 2$) **entonces**

$m_j = \max(m_{inicial}, m_{j-1} - \frac{m_{inicial}}{3})$

si_no

$m_j = \min(m_{j-1} + m_{inicial}, m_{maxima})$

fin_si

si_no

si ($(j - 1) > 5$ and $\frac{\|r_{(j-1)-5}\|}{\|r_{j-1}\|} > 2$) **entonces**

$m_j = \max(m_{inicial}, m_{j-1} - \frac{m_{inicial}}{4})$

si_no

$m_j = \min(m_{j-1} + \frac{m_{inicial}}{2}, m_{maxima})$

fin_si

fin_si

fin_si

si_no

$m_j = m_{j-1}$

fin_si

return [m_j]

Algoritmo 6: Regla de Gonçalez

4.3 Resultados por problema

La regla propuesta a pesar de aumentar el valor del reinicio m consigue un tiempo menor en la mayoría de los casos de prueba. En general, disminuye el valor de m a partir de un valor inicial. Si llega a un valor mínimo, aumenta el valor inicial y nuevamente disminuye m a partir de ese nuevo valor inicial.

4.3.1 Problema 1. Matriz Add20

Área: Diseño de circuitos electrónicos

Tipo: Real no simétrica

Tamaño: 2395×2395

Valores no-ceros: 13151

Lado derecho b: proveído con la matriz

Repositorio: Mercado de Matrices

Nro. de condicionamiento est.: 1.76×10^4

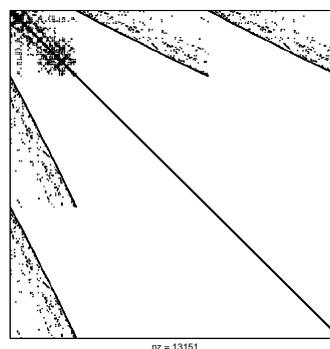


Figura 4.2: Estructura de Add20.

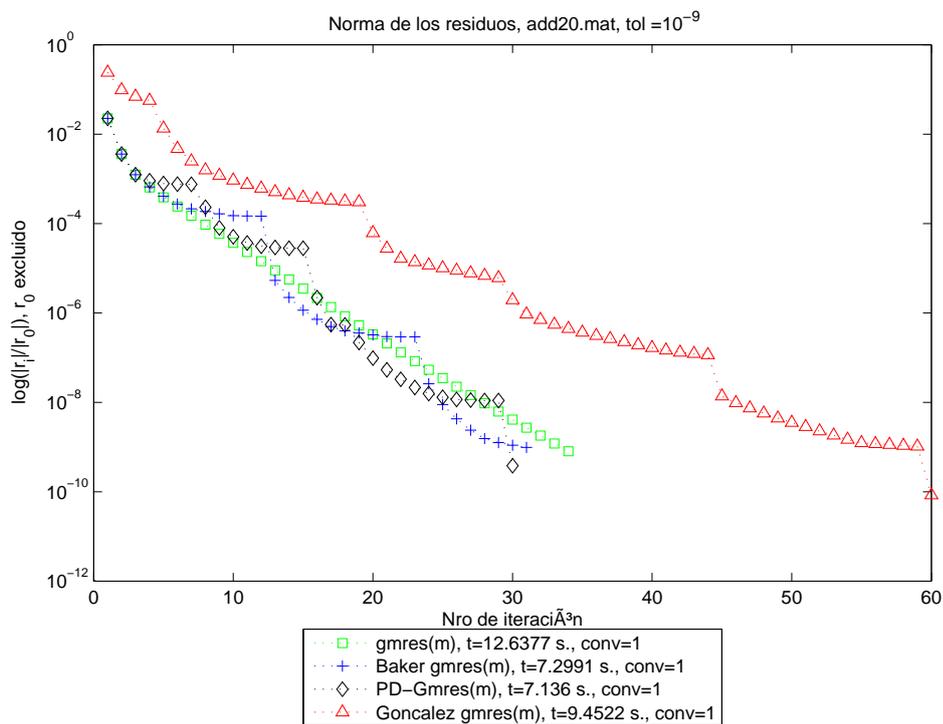


Figura 4.3: Matriz Add20: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

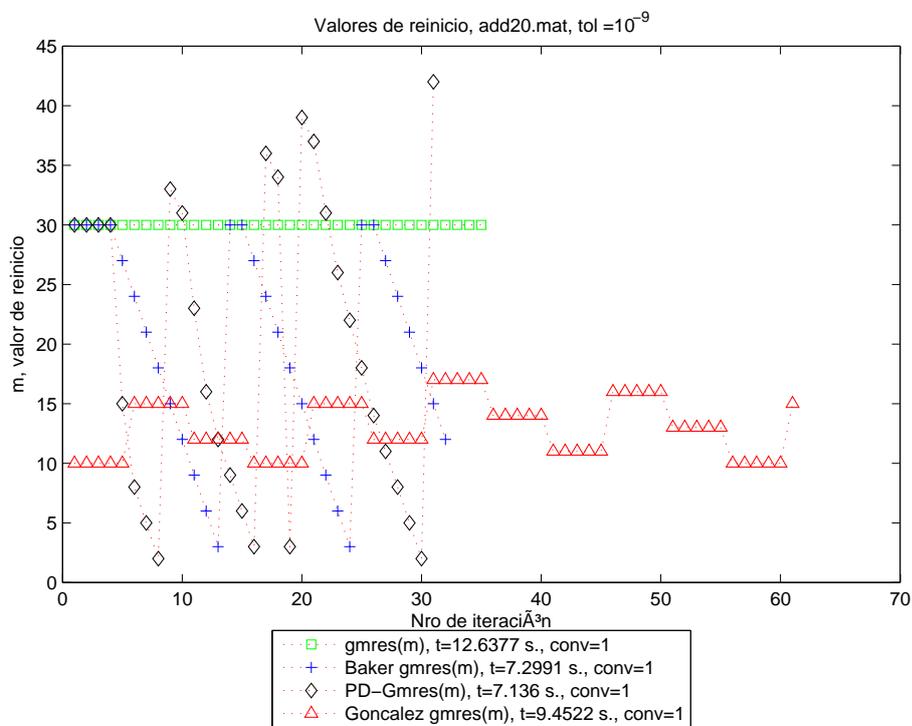


Figura 4.4: Matriz Add20: Reinicios m Vs Iteraciones.

4.3.2 Problema 2. Matriz Cdde1

Área: Dinámica de Fluidos Computacional

Tipo: Real no simétrica

Tamaño: 961×961

Valores no-ceros: 4681

Lado derecho b: generado aleatoriamente

Repositorio: Mercado de Matrices

Nro. de condicionamiento est.: 4.1×10^3

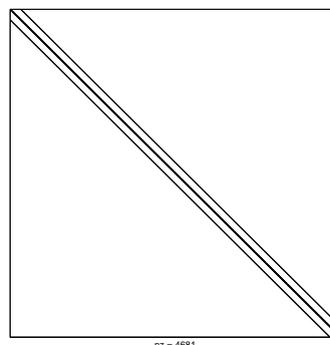


Figura 4.5: Estructura de Cdde1.

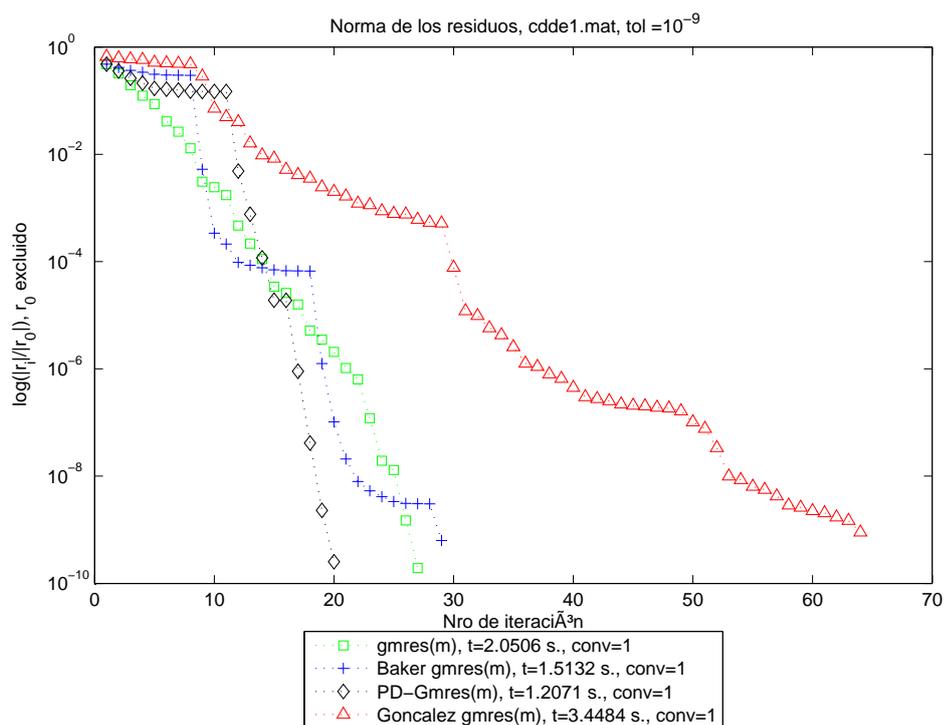


Figura 4.6: Matriz Cdde1: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

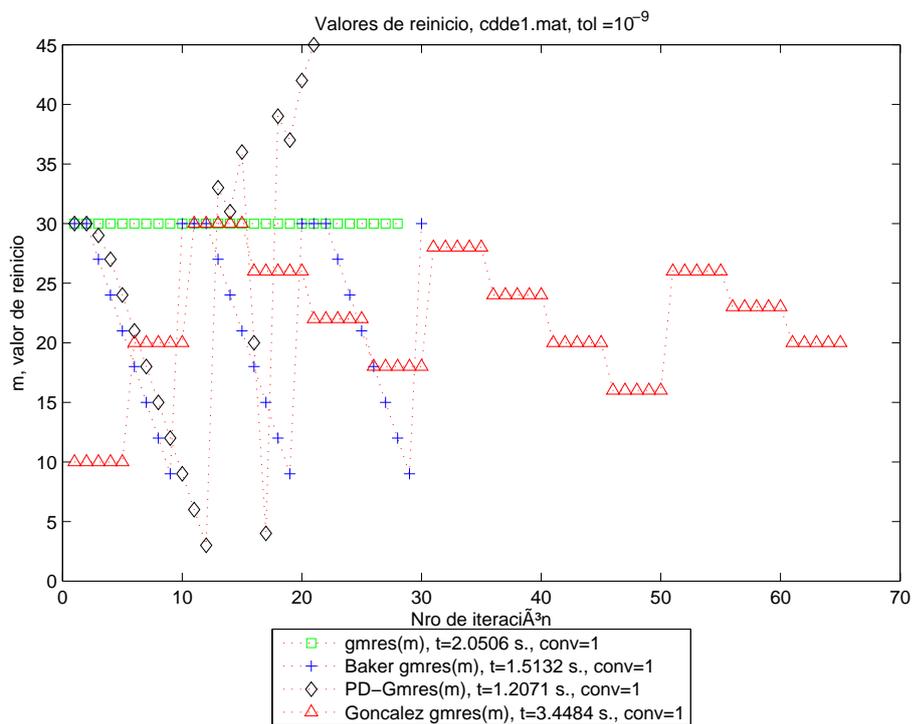


Figura 4.7: Matriz Cdde1: Reinicios m Vs Iteraciones.

4.3.3 Problema 3. Matriz Circuit_2

Área: Simulación de Circuitos

Tipo: Real no simétrica

Tamaño: 4510 × 4510

Valores no-ceros: 21199

Lado derecho b: proveído con la matriz

Repositorio: Universida de Florida

Nro. de condicionamiento est.: 7.12×10^6

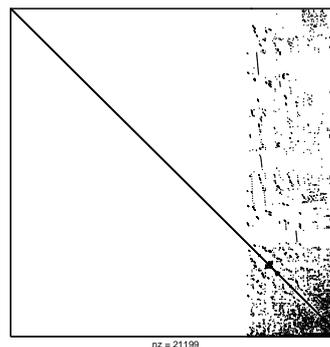


Figura 4.8: Estructura de Circuit_2.

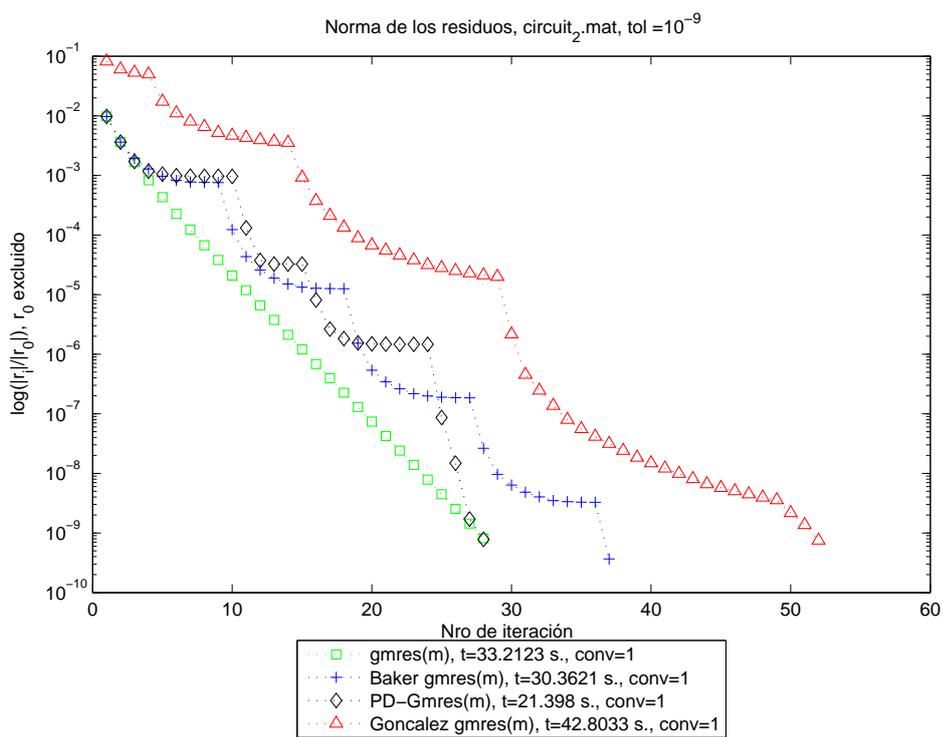


Figura 4.9: Matriz Circuit2: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

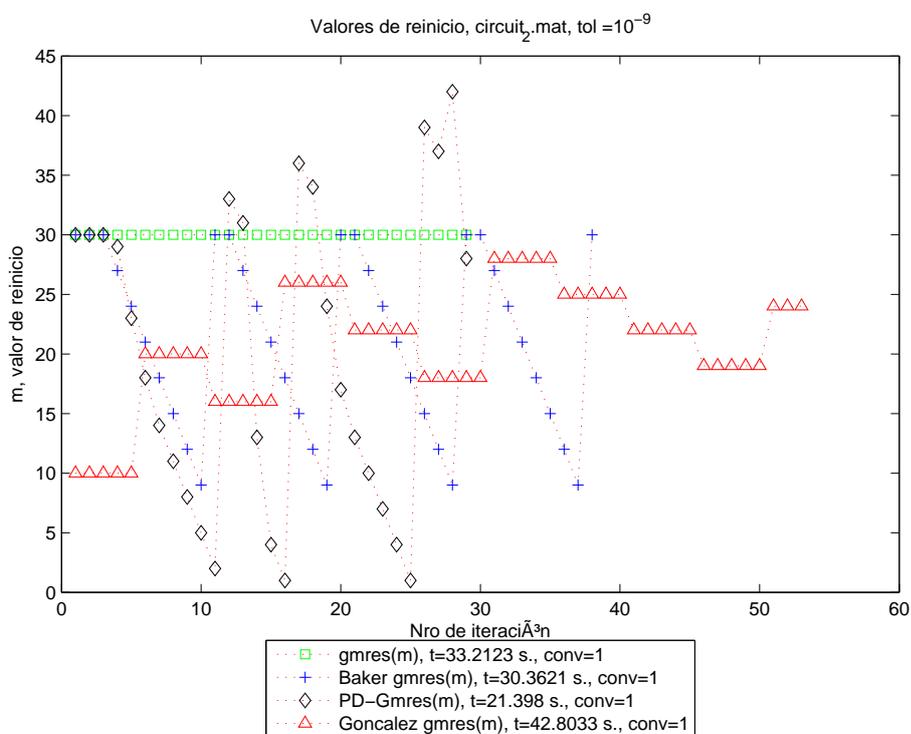


Figura 4.10: Matriz Circuit2: Reinicios m Vs Iteraciones.

4.3.4 Problema 4. Matriz Fpga_trans_01

Área: Simulación de Circuitos

Tipo: Real no simétrica

Tamaño: 1220×1220

Valores no-ceros: 7382

Lado derecho b: proveído con la matriz

Repositorio: Universida de Florida

Nro. de condicionamiento est.: 2.88×10^4

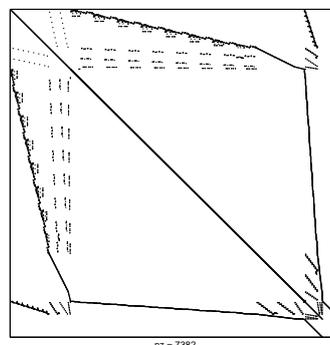


Figura 4.11: Estructura de Fpga_trans_01.

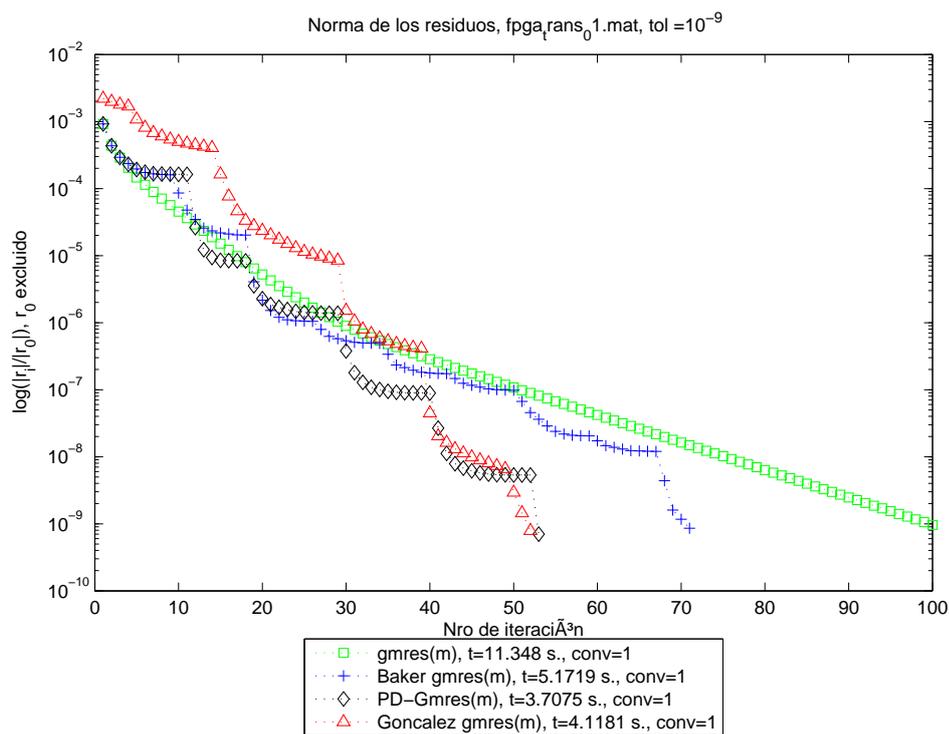


Figura 4.12: Matriz Fpga_trans_01: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

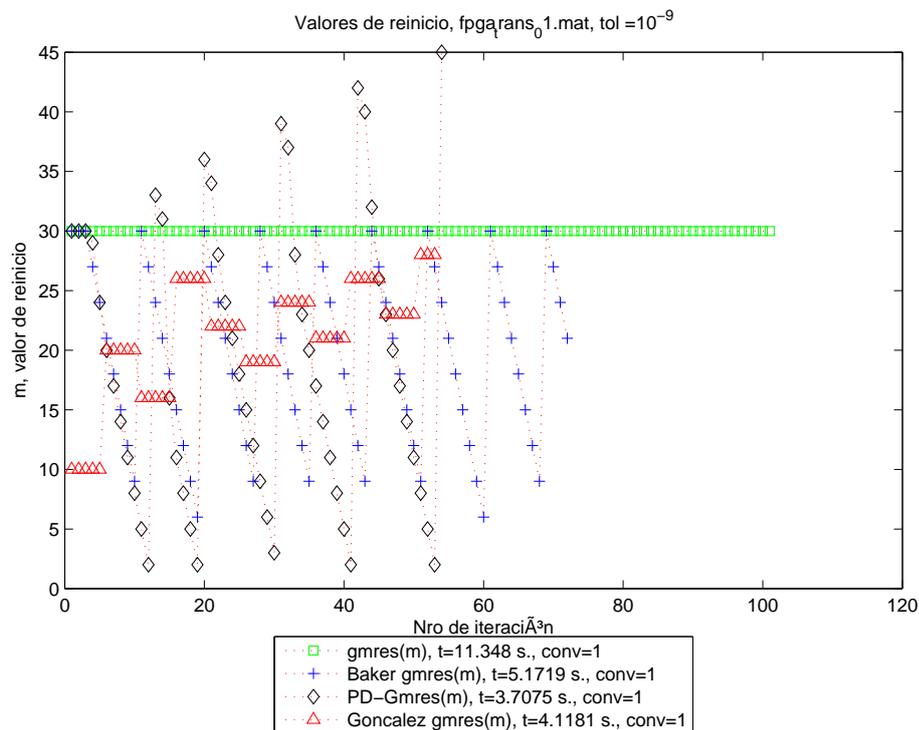


Figura 4.13: Matriz Fpga_trans_01: Reinicios m Vs Iteraciones.

4.3.5 Problema 5. Matriz Orsirr_1

Área: Simulación de Reservorio de Petróleo

Tipo: Real no simétrica

Tamaño: 1030×1030

Valores no-ceros: 6858

Lado derecho b: generado aleatoriamente

Repositorio: Mercado de Matrices

Nro. de condicionamiento est.: 1×10^2

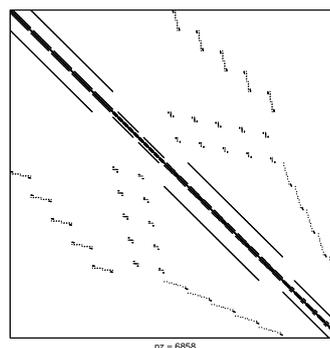


Figura 4.14: Estructura de Orsirr_1.

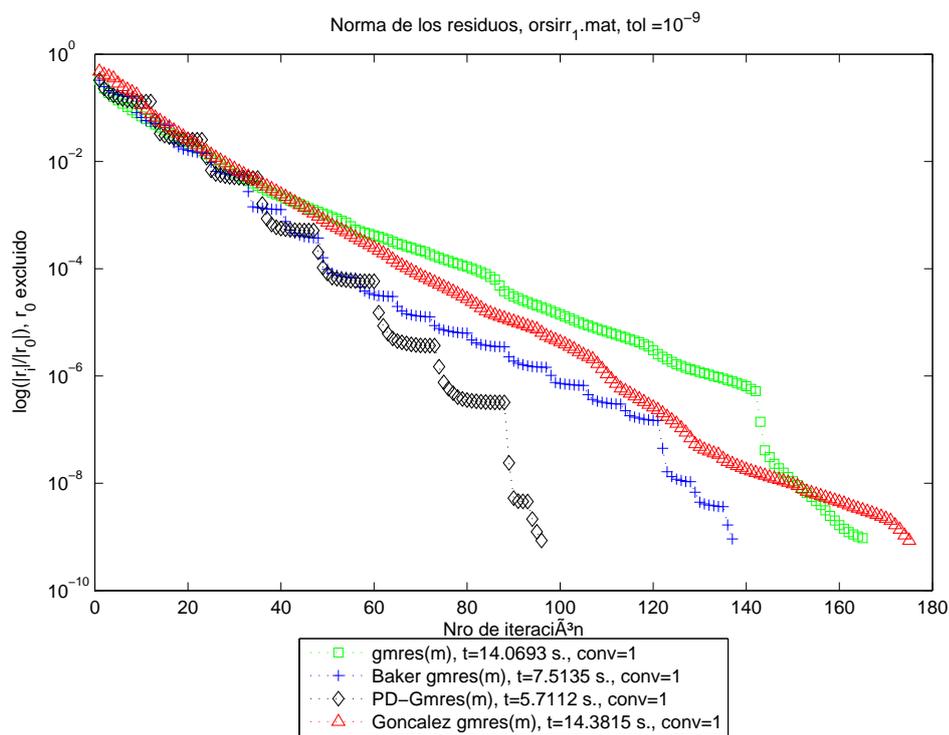


Figura 4.15: Matriz Orsirr_1: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

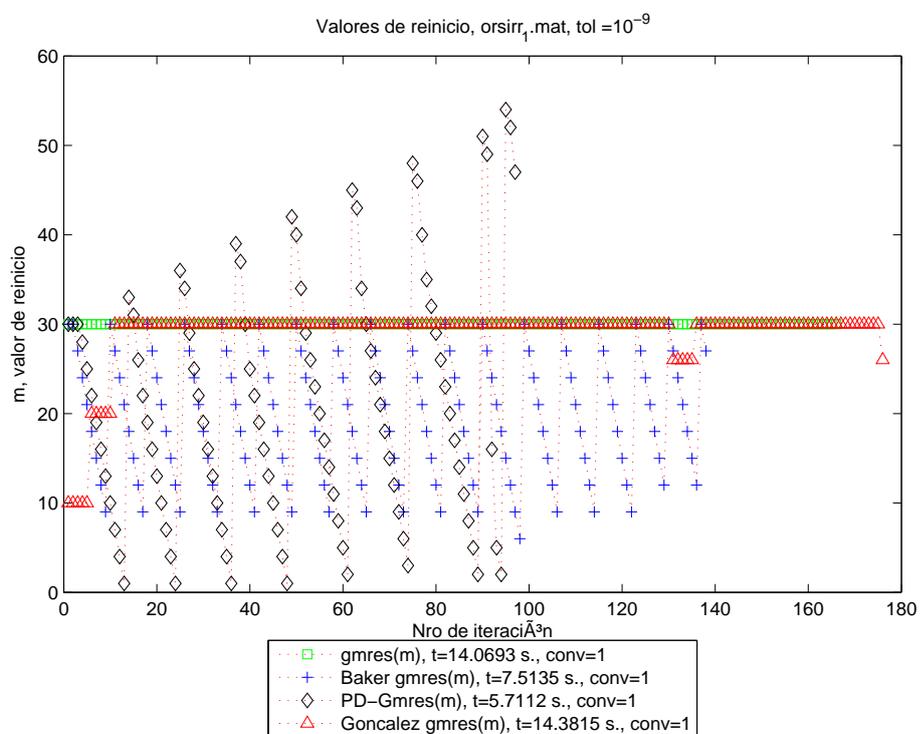


Figura 4.16: Matriz Orsirr_1: Reinicios m Vs Iteraciones.

4.3.6 Problema 6. Matriz Orsreg_1

Área: Simulación de Reservorio de Petróleo

Tipo: Real no simétrica

Tamaño: 2205×2205

Valores no-ceros: 14133

Lado derecho b: generado aleatoriamente

Repositorio: Mercado de Matrices

Nro. de condicionamiento est.: 1×10^2

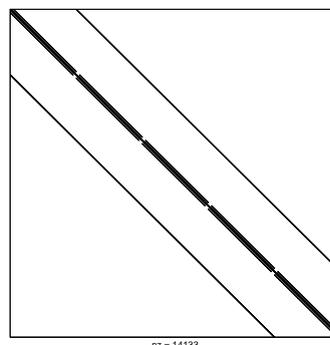


Figura 4.17: Estructura de Orsreg_1.

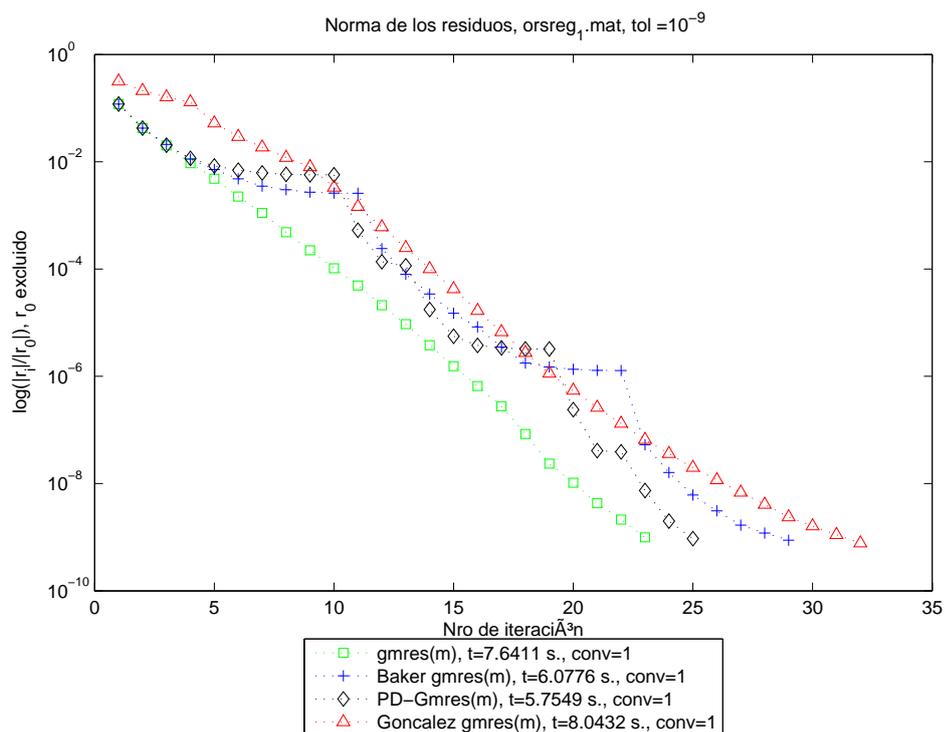


Figura 4.18: Matriz Orsreg_1: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

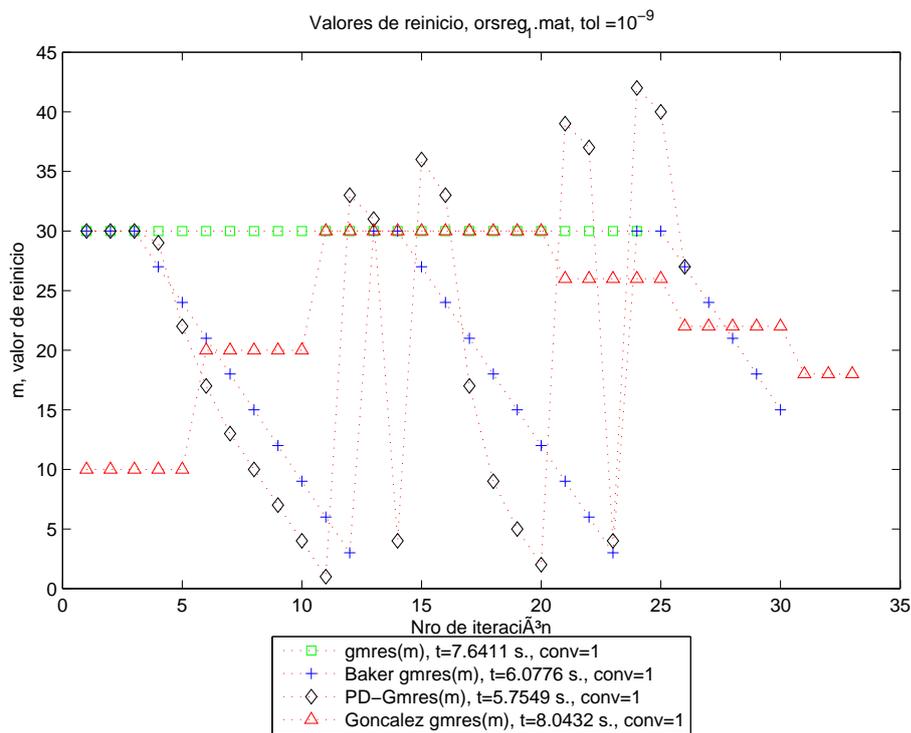


Figura 4.19: Matriz Orsreg_1: Reinicios m Vs Iteraciones.

4.3.7 Problema 7. Matriz Pde2961

Área: Modelo de un problema de EDP

Tipo: Real no simétrica

Tamaño: 2961×2961

Valores no-ceros: 14585

Lado derecho b: generado aleatoriamente

Repositorio: Mercado de Matrices

Nro. de condicionamiento est.: 9.49×10^2

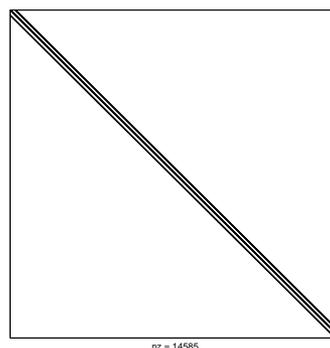


Figura 4.20: Estructura de Pde2961.

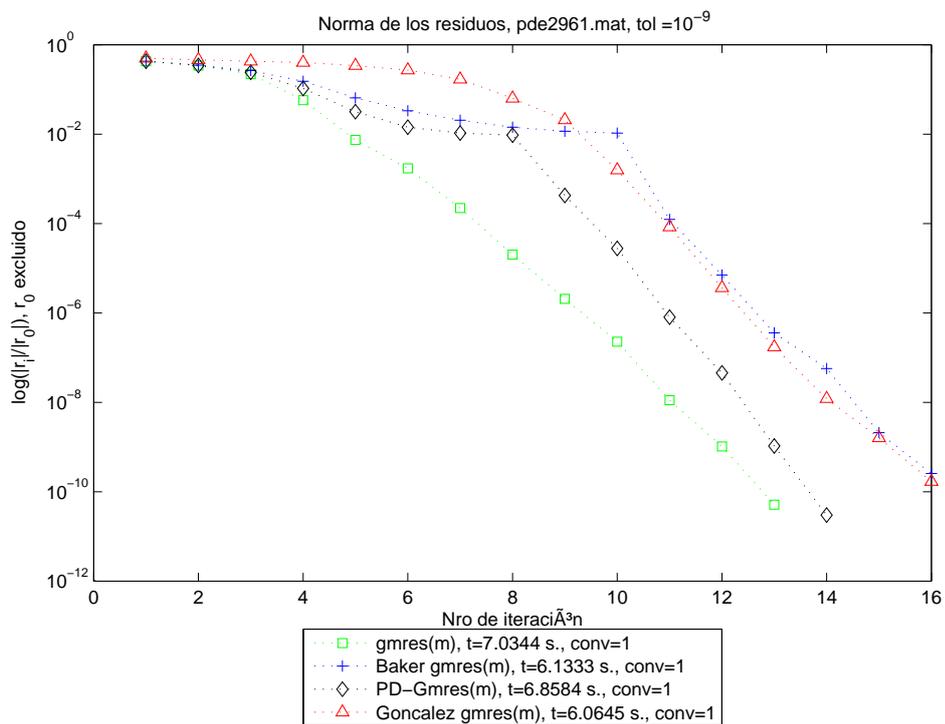


Figura 4.21: Matriz Pde2961: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

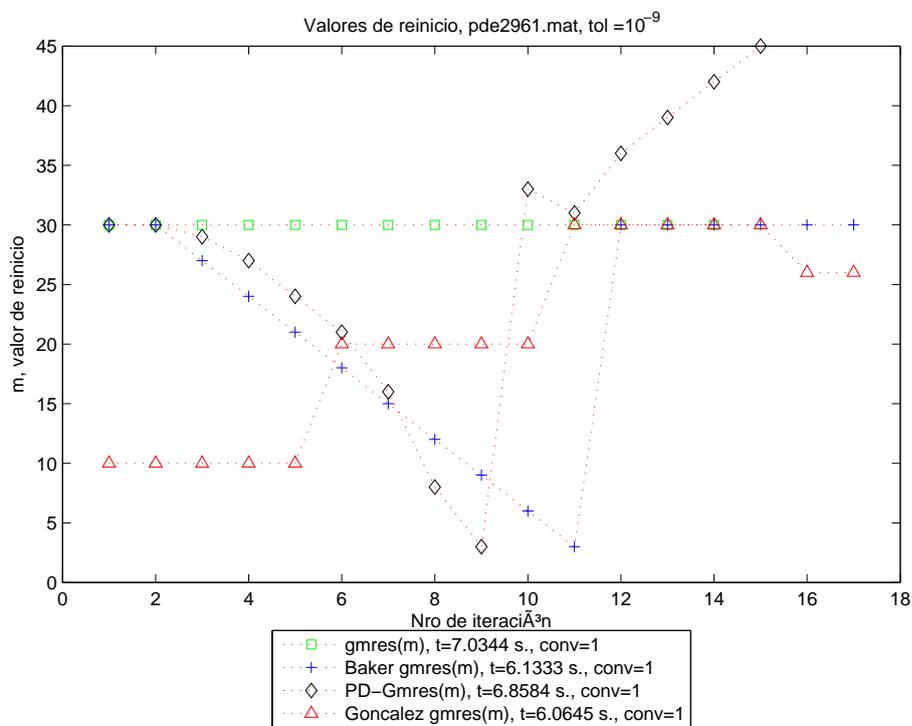


Figura 4.22: Matriz Pde2961: Reinicios m Vs Iteraciones.

4.3.8 Problema 8. Matriz Raefsky1

Área: Flujo de Fluido Incompresible
 Tipo: Real no simétrica
 Tamaño: 3242×3242
 Valores no-ceros: 294276
 Lado derecho b: proveído con la matriz
 Repositorio: Universidad de Florida
 Nro. de condicionamiento est.: 3.16×10^4

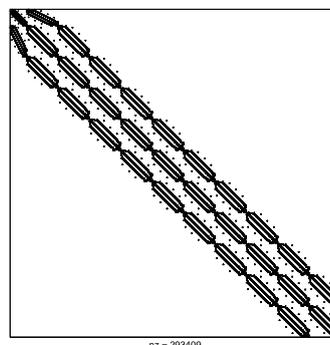


Figura 4.23: Estructura de Raefsky1.

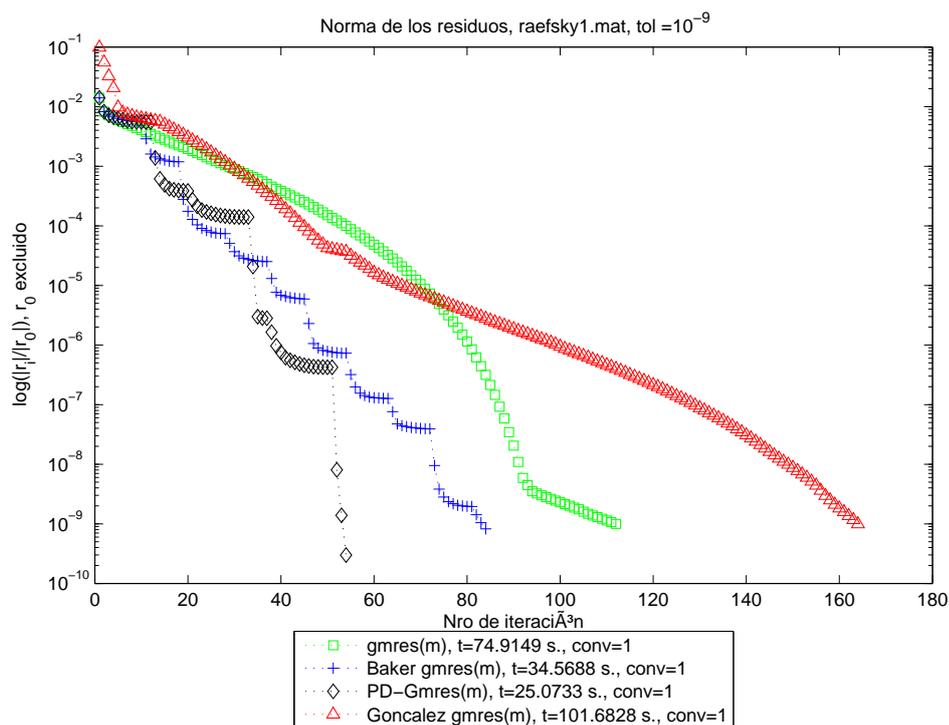


Figura 4.24: Matriz Raefsky1: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

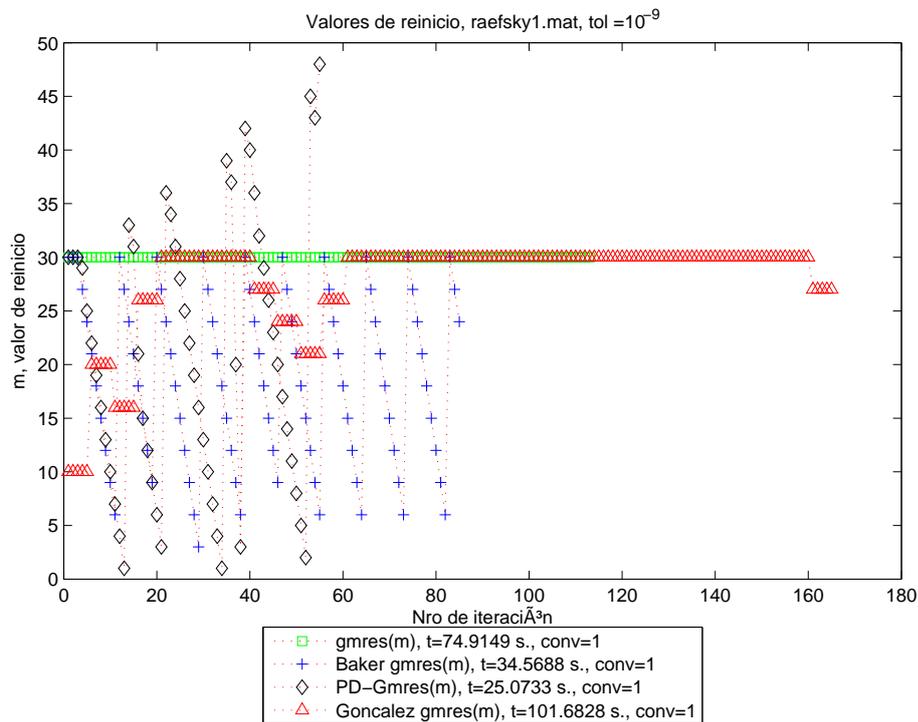


Figura 4.25: Matriz Raefsky1: Reinicios m Vs Iteraciones.

4.3.9 Problema 9. Matriz Raefsky2

Área: Flujo de Fluido Incompresible

Tipo: Real no simétrica

Tamaño: 3242×3242

Valores no-ceros: 293551

Lado derecho b: proveído con la matriz

Repositorio: Universidad de Florida

Nro. de condicionamiento est.: 1.08×10^4

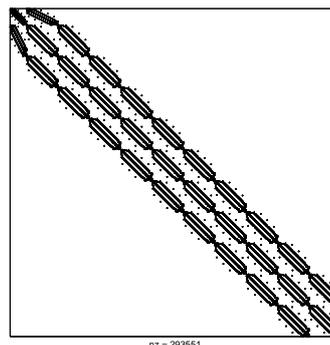


Figura 4.26: Estructura de Raefsky2.

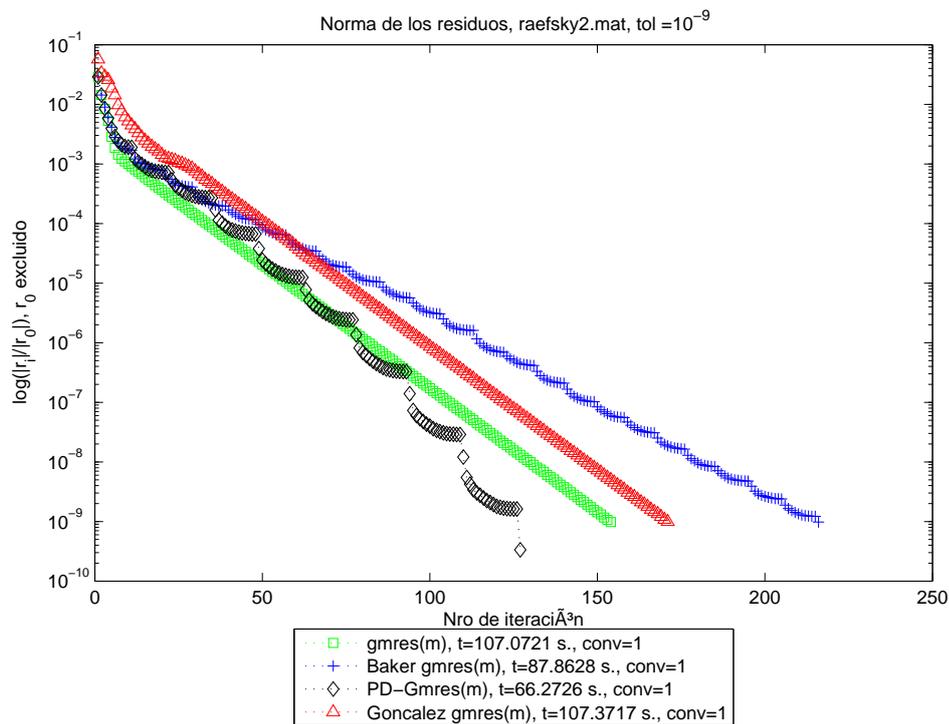


Figura 4.27: Matriz Raefsky2: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

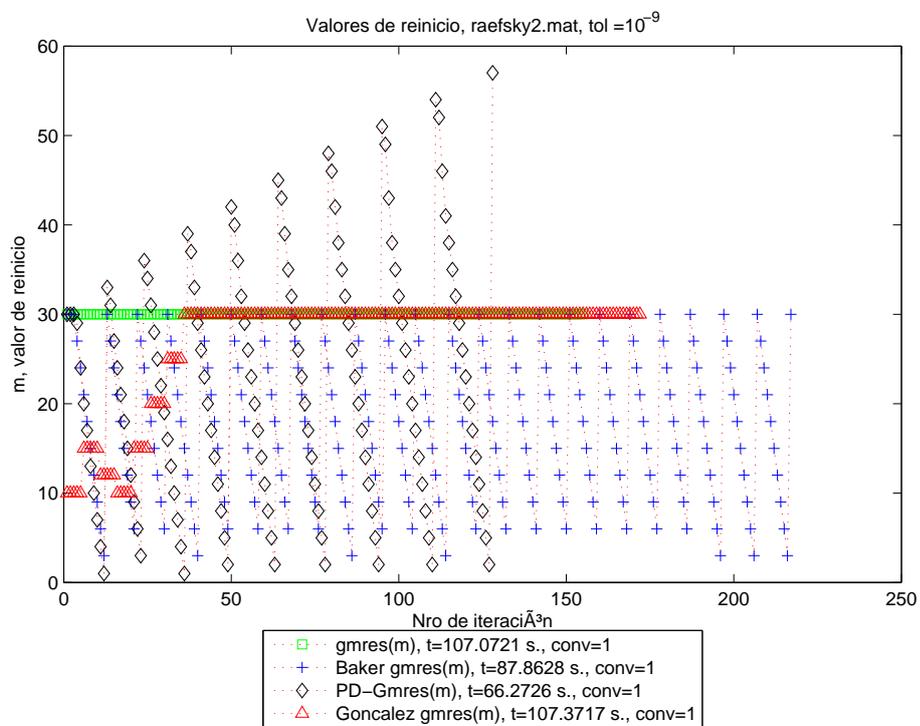


Figura 4.28: Matriz Raefsky2: Reinicios m Vs Iteraciones.

4.3.10 Problema 10. Matriz Rdb2048

Área: Modelo Brusselator para Reacción Difusión

Tipo: Real no simétrica

Tamaño: 2048×2048

Valores no-ceros: 12032

Lado derecho b: generado aleatoriamente

Repositorio: Mercado de Matrices

Nro. de condicionamiento est.: 6.21×10^3

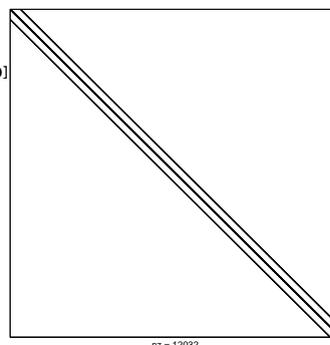


Figura 4.29: Estructura de Rdb2048.

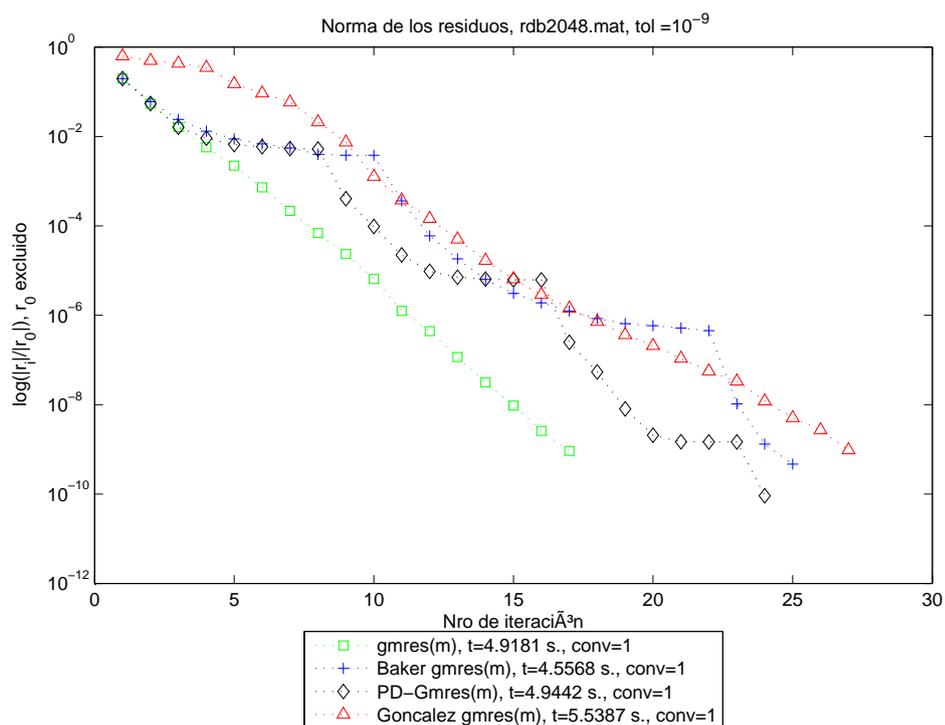


Figura 4.30: Matriz Rdb2048: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

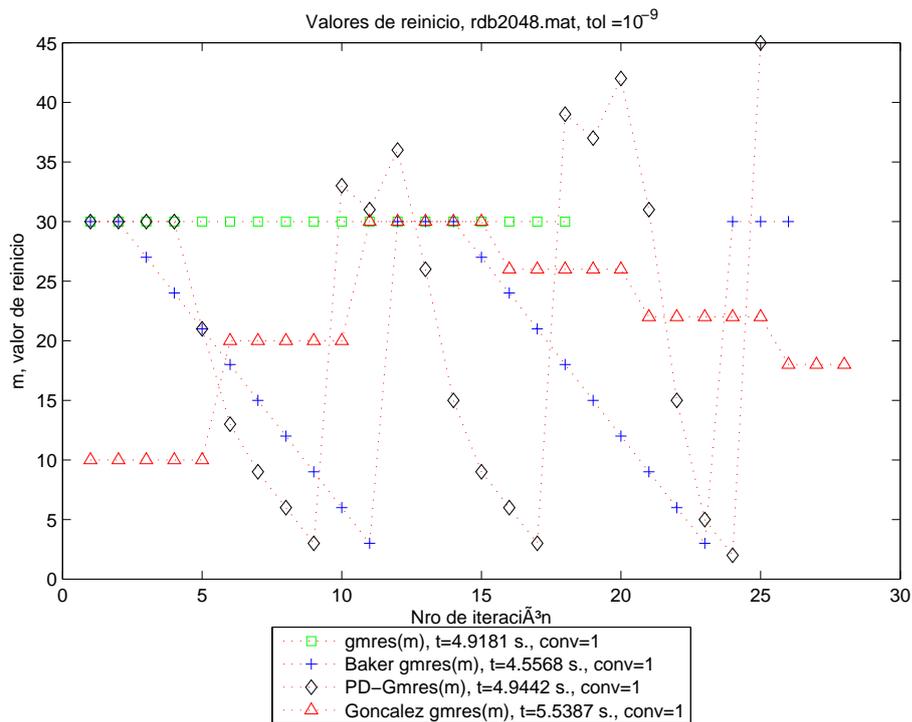


Figura 4.31: Matriz Rdb2048: Reinicios m Vs Iteraciones.

4.3.11 Problema 11. Matriz Sherman4

Área: Simulación de Reservoirio de Petróleo

Tipo: Real no simétrica

Tamaño: 1104×1104

Valores no-ceros: 3786

Lado derecho b: proveído con la matriz

Repositorio: Mercado de Matrices

Nro. de condicionamiento est.: 7.2×10^3

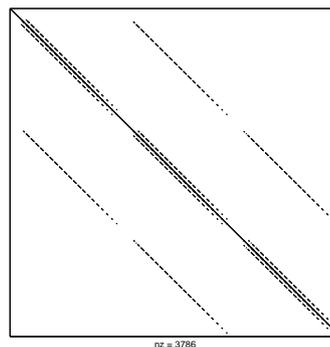


Figura 4.32: Estructura de Sherman4.

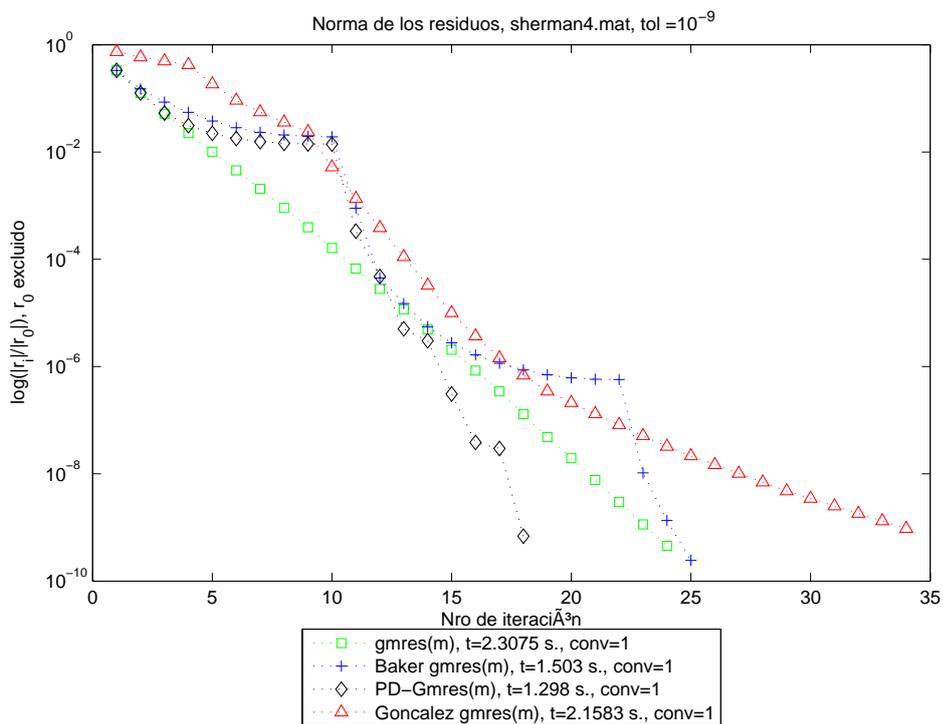


Figura 4.33: Matriz Sherman4: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

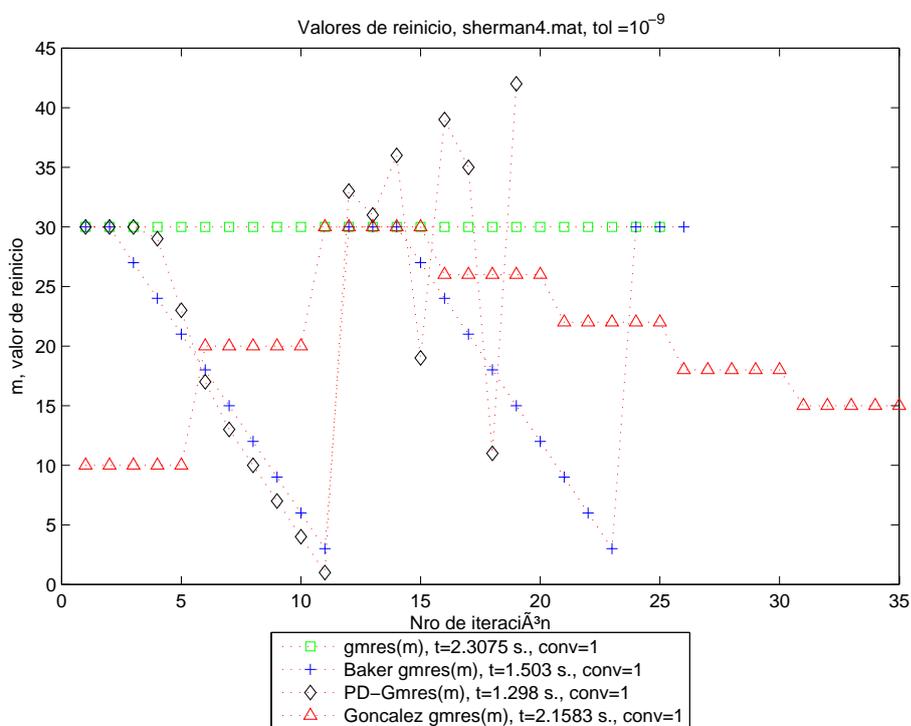


Figura 4.34: Matriz Sherman4: Reinicios m Vs Iteraciones.

4.3.12 Problema 12. Matriz Steam2

Área: Vapor inyectado para recuperar petróleo

Tipo: Real no simétrica

Tamaño: 600×600

Valores no-ceros: 13760

Lado derecho b: generado aleatoriamente

Repositorio: Mercado de Matrices

Nro. de condicionamiento est.: 3.5×10^6

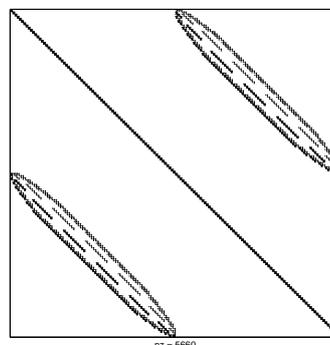


Figura 4.35: Estructura de Steam2.

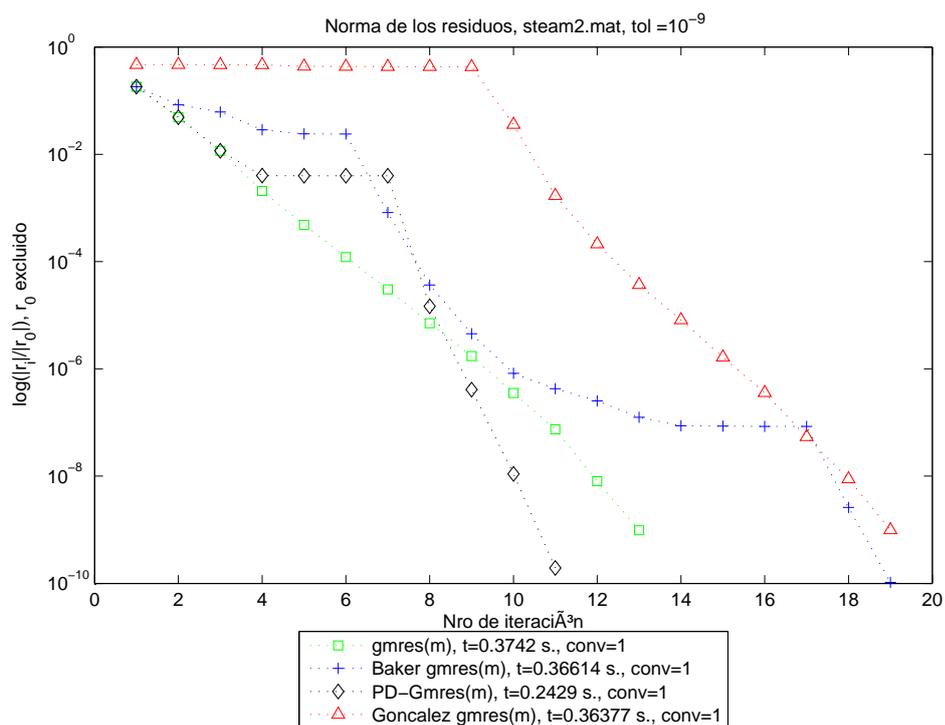


Figura 4.36: Matriz Steam2: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

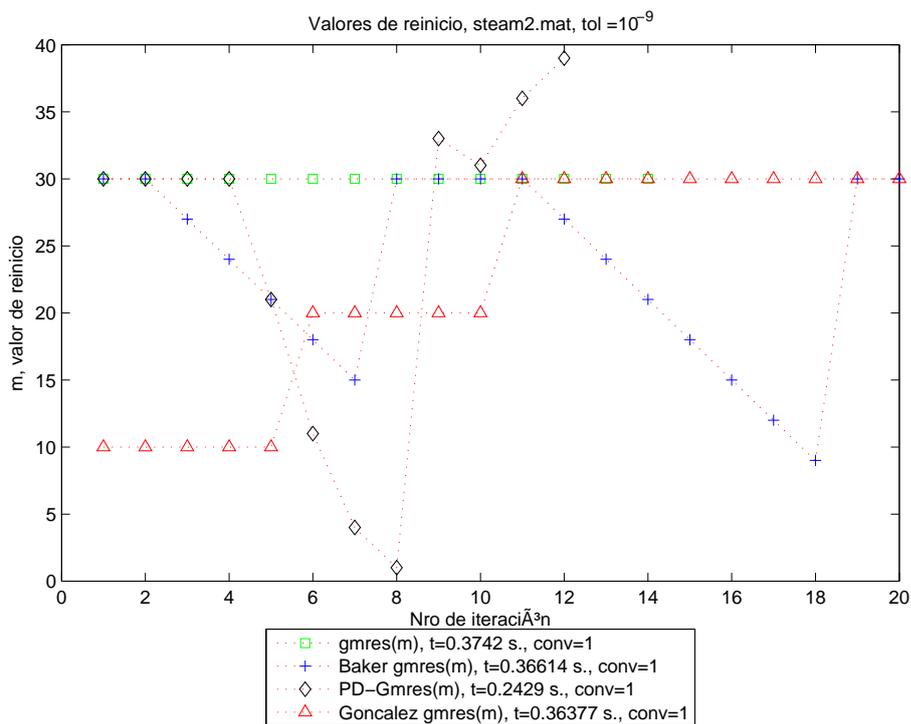


Figura 4.37: Matriz Steam2: Reinicios m Vs Iteraciones.

4.3.13 Problema 13. Matriz Wang2

Área: Ecuaciones de Continuidad de Electrones.

Tipo: Real no simétrica

Tamaño: 2903×2903

Valores no-ceros: 19093

Lado derecho b: generado aleatoriamente

Repositorio: Universidad de Florida

Nro. de condicionamiento est.: 3.18×10^4

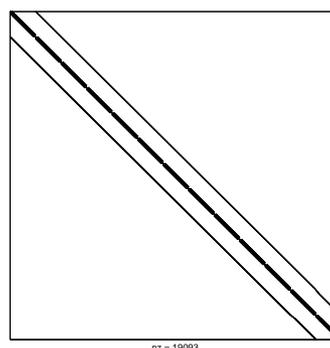


Figura 4.38: Estructura de Wang2.

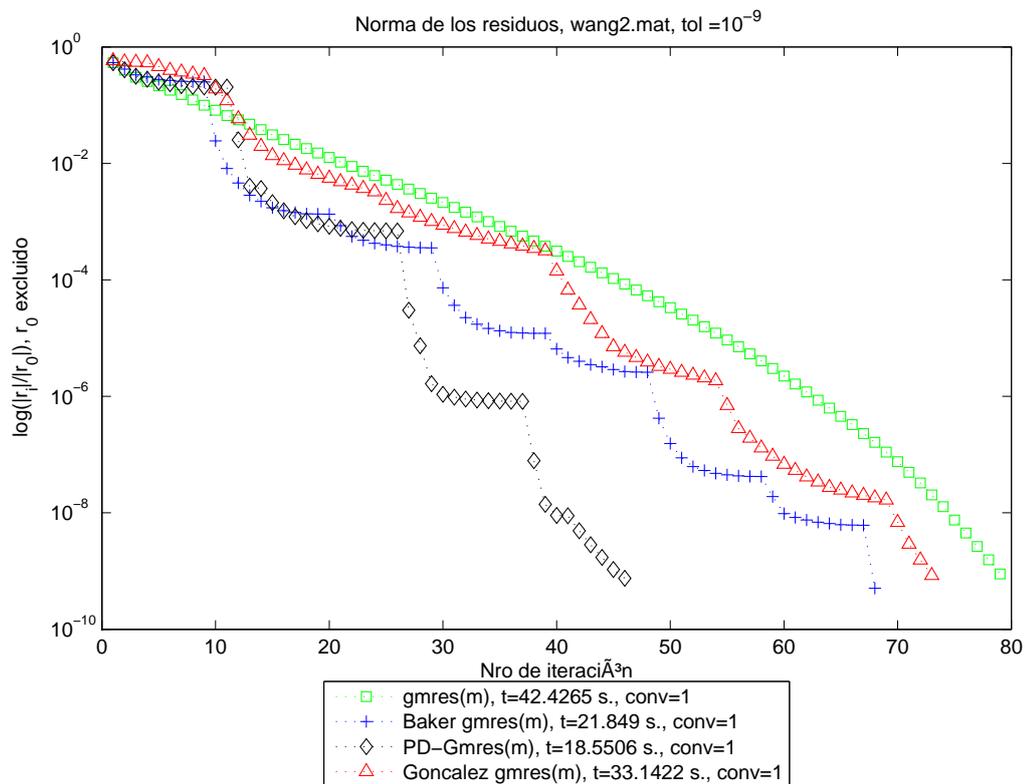


Figura 4.39: Matriz Wang2: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

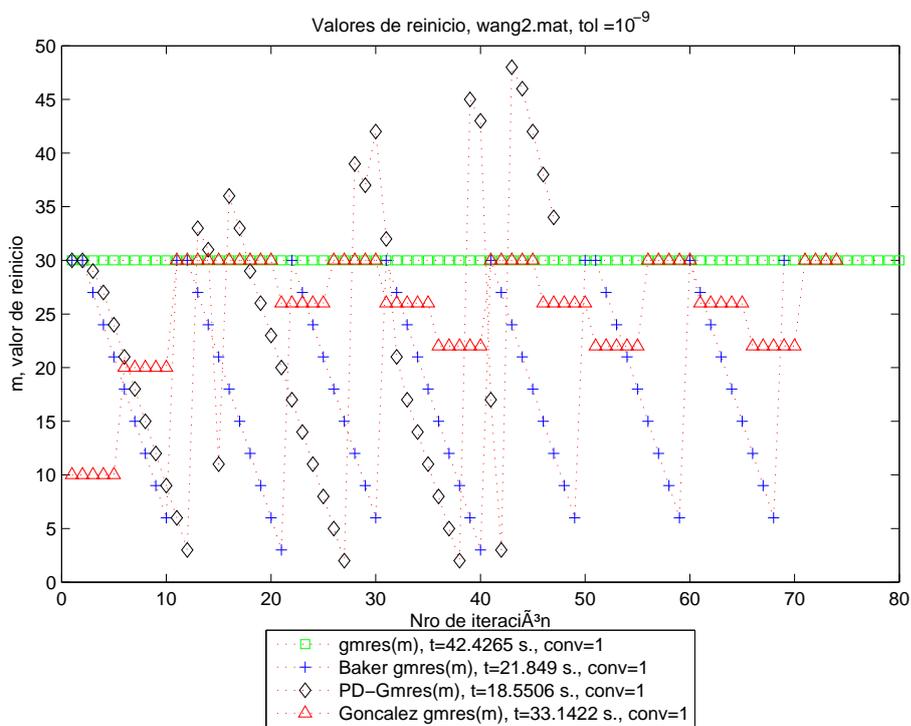


Figura 4.40: Matriz Wang2: Reinicios m Vs Iteraciones.

4.3.14 Problema 14. Matriz Watt_1

Área: Ingeniería de Petróleo.

Tipo: Real no simétrica

Tamaño: 1856×1856

Valores no-ceros: 11360

Lado derecho b: generado aleatoriamente

Repositorio: Mercado de Matrices

Nro. de condicionamiento est.: 5.38×10^9

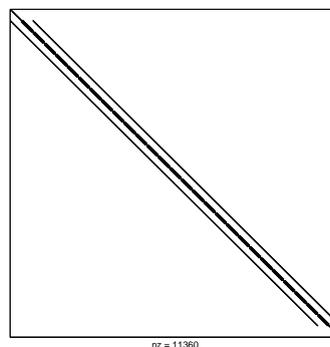


Figura 4.41: Estructura de Watt_1.

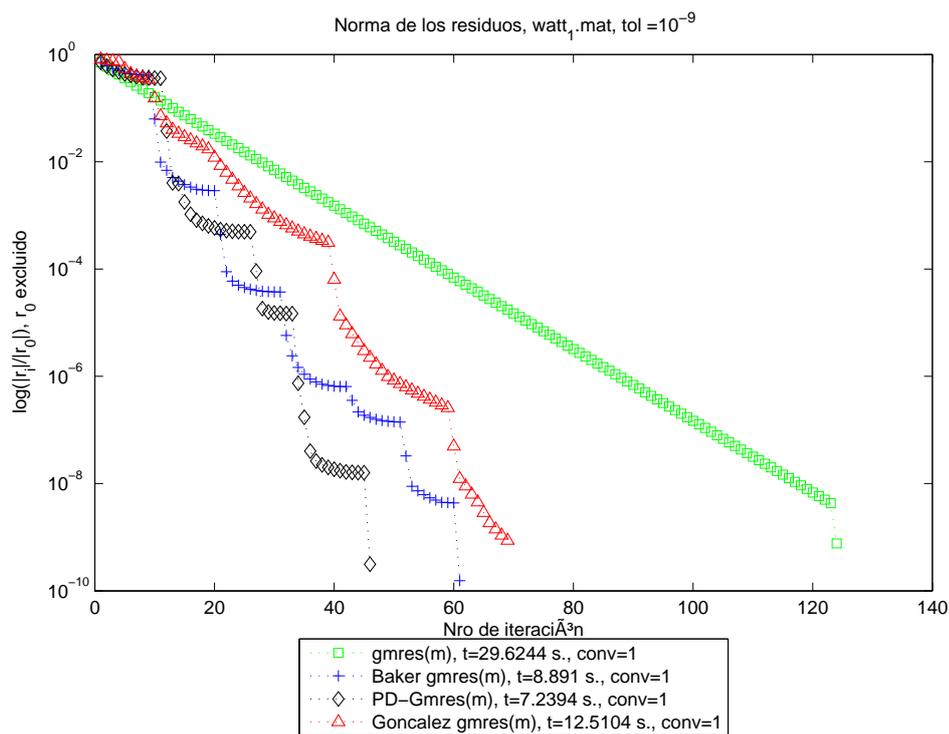


Figura 4.42: Matriz Watt_1: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

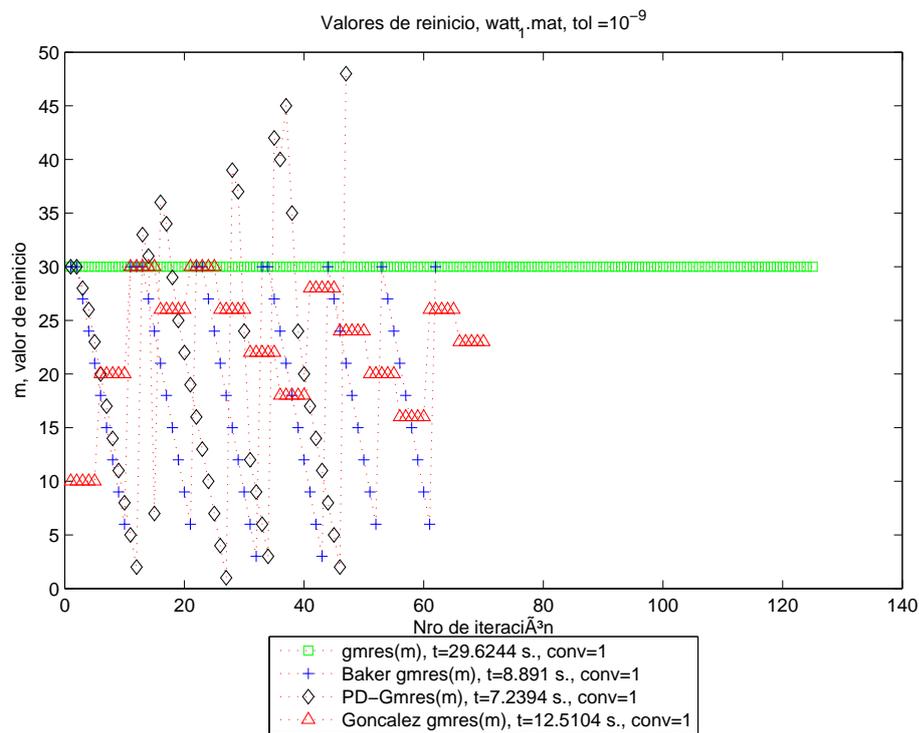


Figura 4.43: Matriz Watt₁: Reinicios m Vs Iteraciones.

4.3.15 Problema 15. Matriz Young3c

Área: Dispersión Acústica

Tipo: Real no simétrica

Tamaño: 841 × 841

Valores no-ceros: 3988

Lado derecho b: generado aleatoriamente

Repositorio: Mercado de Matrices

Nro. de condicionamiento est.: 1.8×10^4

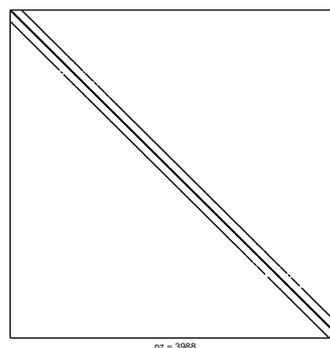


Figura 4.44: Estructura de Young3c.

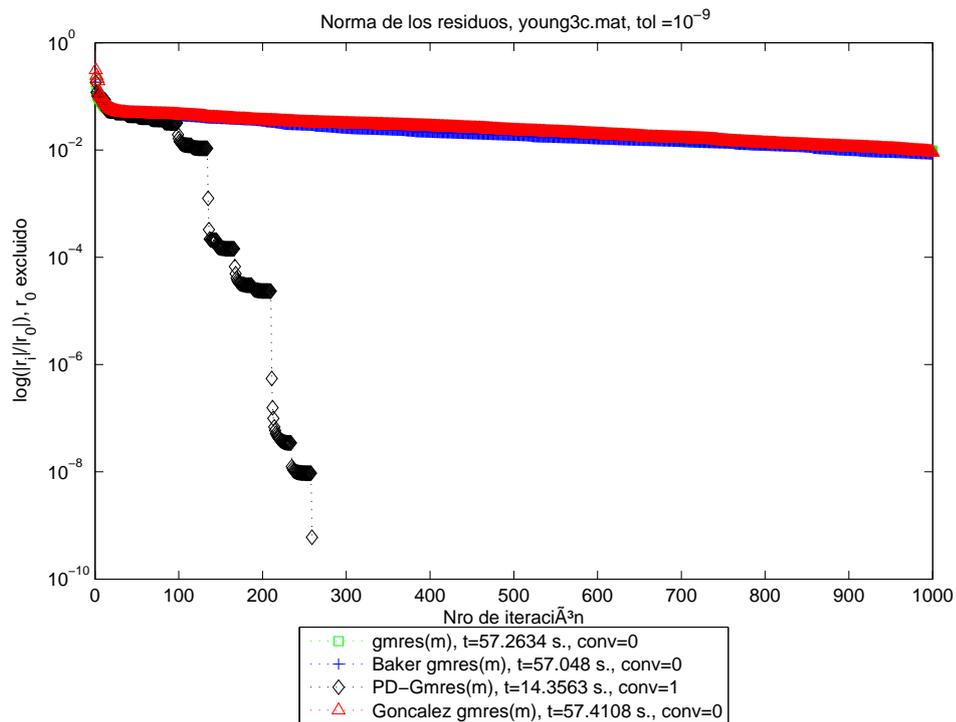


Figura 4.45: Matriz Young3c: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

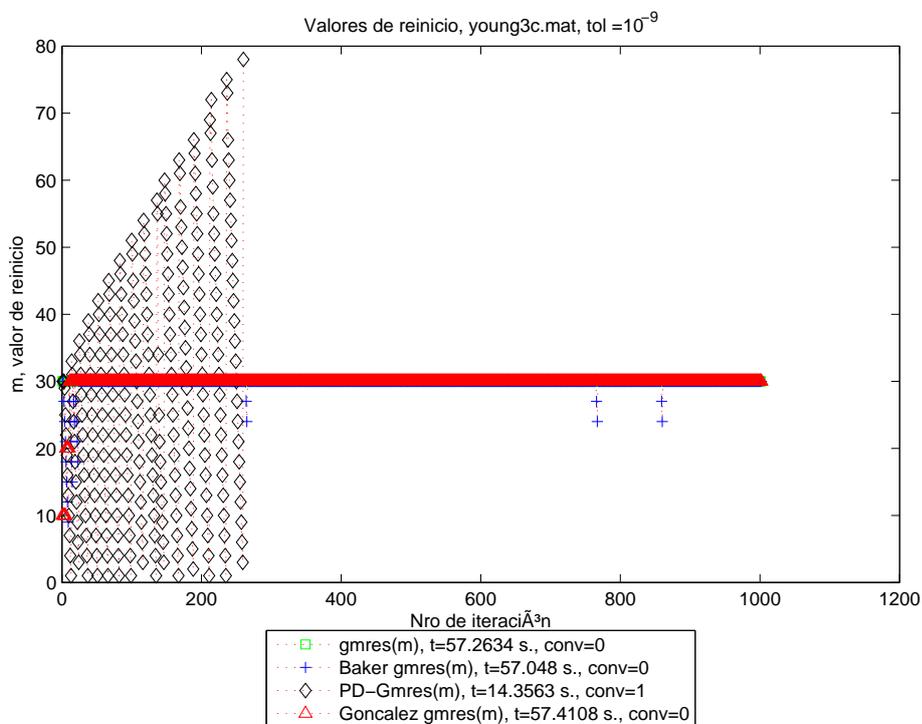


Figura 4.46: Matriz Young3c: Reinicios m Vs Iteraciones.

4.4 Consideraciones sobre estancamiento

Un método iterativo se estanca cuando $\|r_j\| \approx \|r_{j+1}\|$ para todo $j > j_0$. Es decir, cuando la norma del residuo prácticamente no varía. Por otra parte, existen problemas con convergencia lenta. En ellos, la norma del residuo decrece pero muy lentamente.

Para evitar estancamiento y convergencia lenta, [GRLB08] propusieron un método, que llamamos Hybrid-GMRES. Detectan el problema de convergencia utilizando el coseno entre dos residuos, porque también observaron residuos aproximadamente paralelos en los experimentos que realizaron. Hybrid-GMRES resultó mejor que Deflated GMRES para los ejemplos presentados en [GRLB08].

En [GRLB08] el parámetro m es fijo, pero cuando encuentra estancamiento, cambia el residuo para el siguiente reinicio. Es decir, usa un residuo que no proviene directamente de GMRES, por eso lo llamamos residuo híbrido. Éste se obtiene usando una combinación lineal de los dos últimos residuos r_j y r_{j-1} . Es decir, $r_h = \alpha r_{j-1} + (1 - \alpha)r_j$ donde $\alpha = -\frac{\langle r_j, r_{j-1} - r_j \rangle}{\langle r_{j-1} - r_j, r_{j-1} - r_j \rangle}$ se obtuvo minimizando la norma $\|r_h\|$ (ver [Bre98]).

A continuación comparamos PD-GMRES, la regla propuesta en este trabajo, con Hybrid-GMRES. Hybrid-GMRES no fue incluido en las comparaciones anteriores porque es en general más lento que PD-GMRES, la regla de Baker o la regla de Gonçalves. Es más lento porque el cálculo del residuo híbrido incluye operaciones vectoriales, mientras que las reglas que varían m utilizan la norma del residuo, que es un escalar. La comparación es cualitativa para observar si los métodos superan el estancamiento o la convergencia lenta y encuentran la solución aproximada según la tolerancia indicada.

4.4.1 Experimento de Convergencia Lenta

Como ejemplo utilizamos el Problema 15 (4.44). Solamente PD-GMRES convergió para el Problema de Prueba 15 (matriz Young3c). Note que el residuo híbrido reduce el residuo pero su comportamiento para este problema no es muy diferente del comportamiento mostrado por GMRES(m). Ver la Figura 4.4.1.

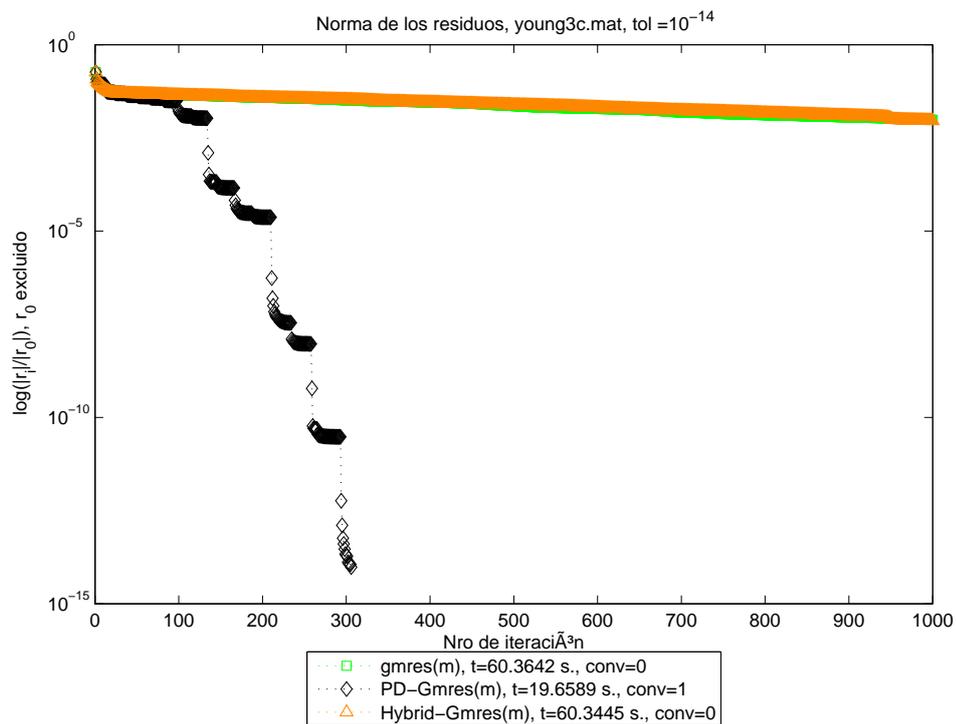


Figura 4.47: Matriz Young3c: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

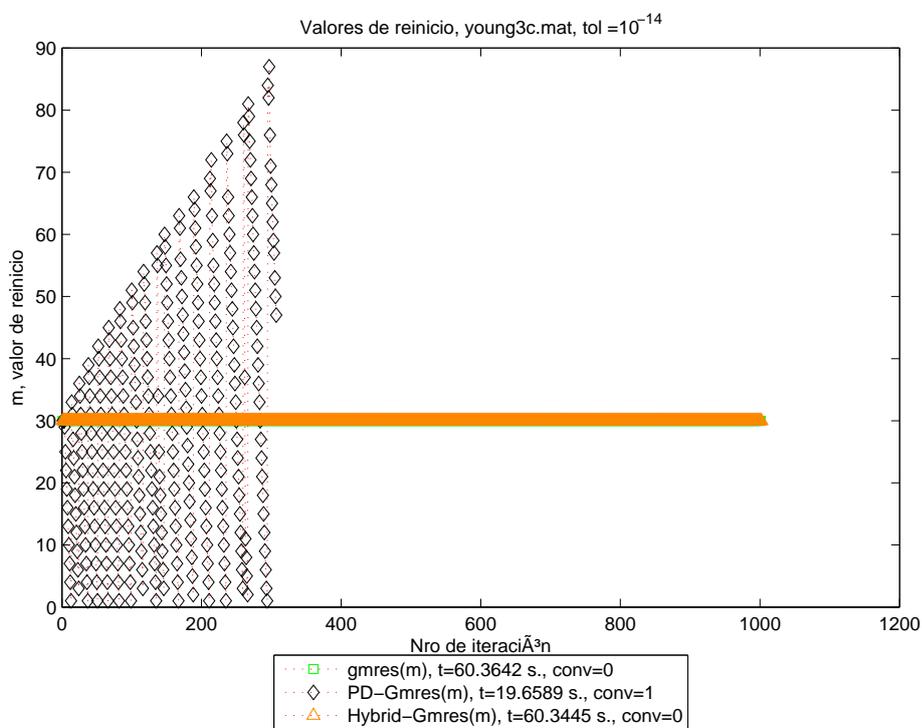


Figura 4.48: Matriz Young3c: Reinicios m Vs Iteraciones.

4.4.2 Experimentos de Estancamiento

Debemos hacer notar que, aunque la reducción de la norma residual relativa de PD-GMRES es mejor que la de Hybrid-GMRES, ninguno de los métodos converge bajo la tolerancia indicada (10^{-14}). Para casos de estancamiento utilizamos las matrices Mahindas y Shl400. Ambas provenientes del Mercado de Matrices. Si bien la regla que proponemos reduce el residuo, la reducción es poca. Ver las figuras (4.4.2,4.4.2).

Matriz Mahindas

Área: Mod. Económico de Victoria (AU 1880)

Tipo: Real no simétrica

Tamaño: 1258×1258

Valores no-ceros: 7682

Lado derecho b: proveído

Repositorio: Mercado de Matrices

Nro. de condicionamiento est.: 1.034×10^{13}

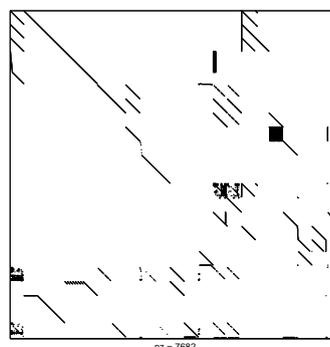


Figura 4.49: Estructura de Mahindas.

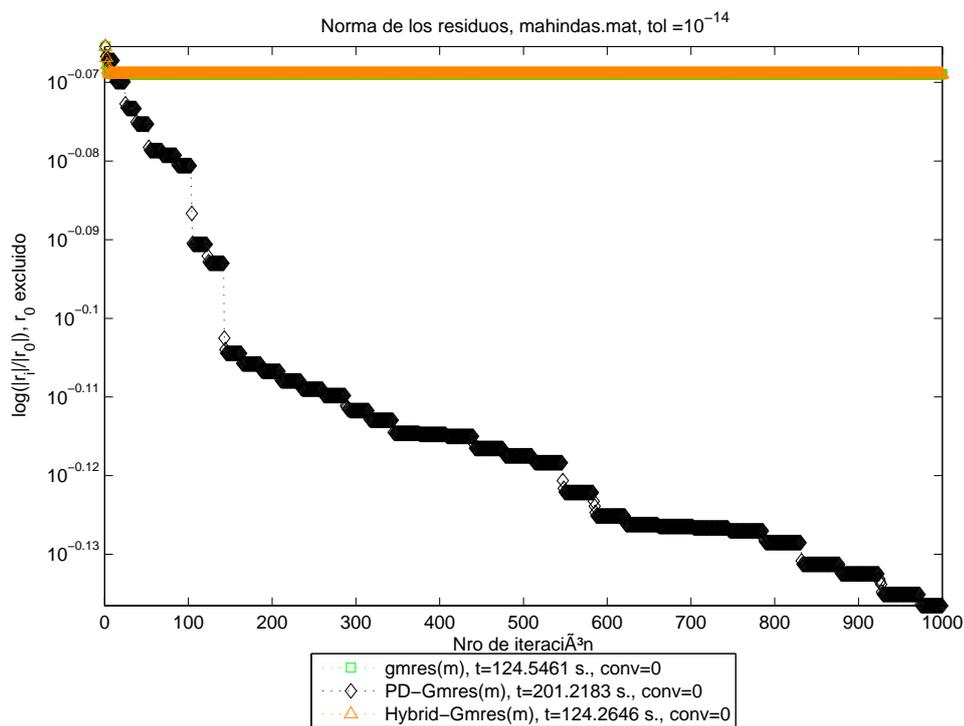


Figura 4.50: Matriz Mahindas: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

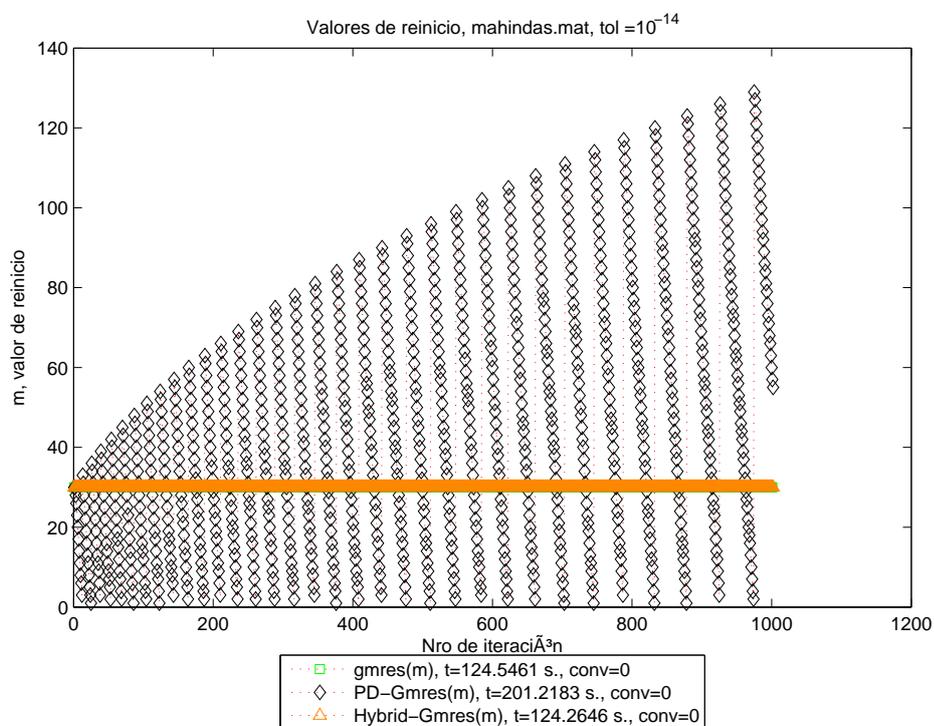


Figura 4.51: Matriz Mahindas: Reinicios m Vs Iteraciones.

Matriz Shl_400

Área: Matriz básica de un Método Simplex

Tipo: Real no simétrica

Tamaño: 663×663

Valores no-ceros: 1712

Lado derecho b: generado aleatoriamente

Repositorio: Mercado de Matrices

Nro. de condicionamiento est.: 1.92×10^7

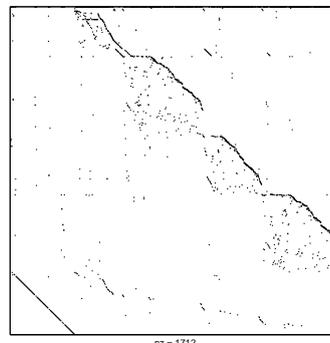


Figura 4.52: Estructura de Shl_400.

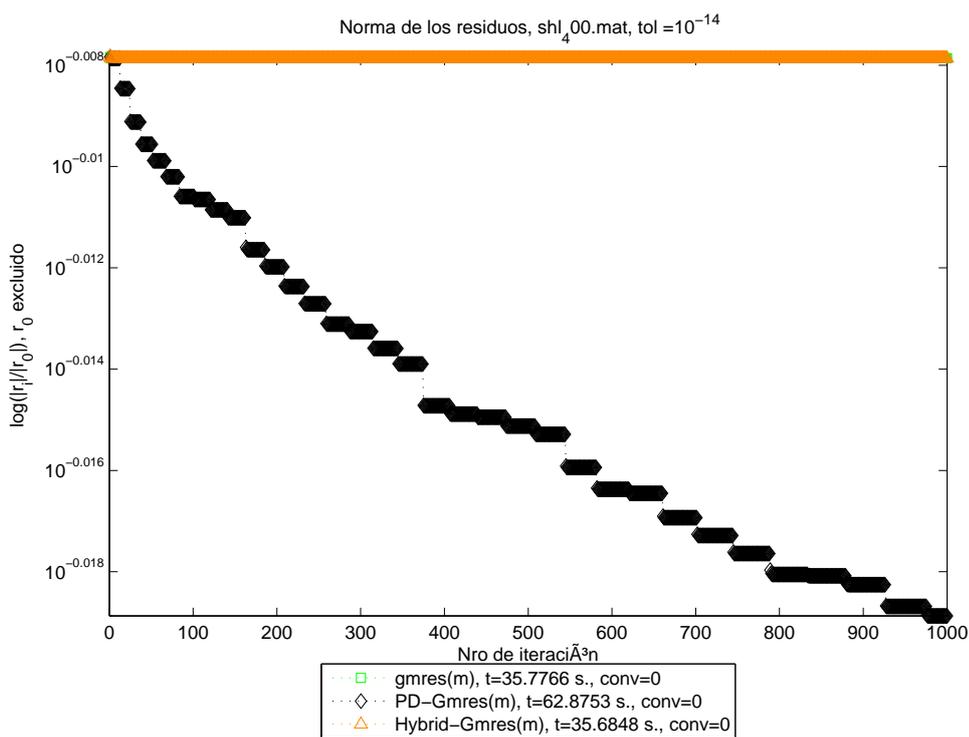


Figura 4.53: Matriz Shl_400: $\log(\|r_j\| / \|r_0\|)$ Vs Iteraciones

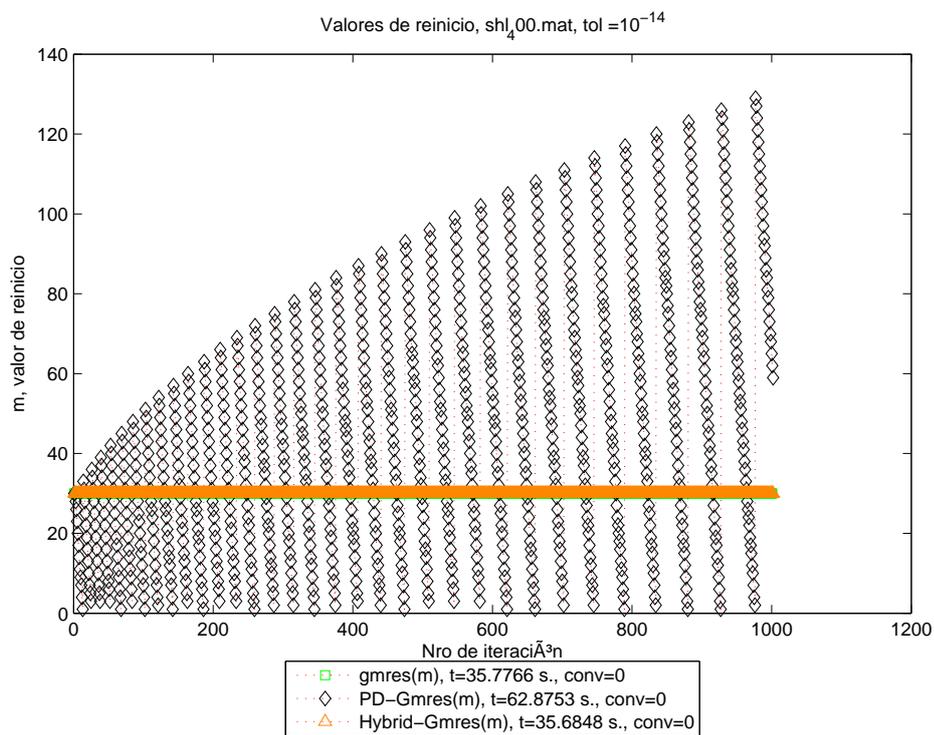


Figura 4.54: Matriz Shl_400: Reinicios m Vs Iteraciones.

Capítulo 5

CONCLUSIONES

Este trabajo exploró el cambio del parámetro de reinicio m del método iterativo GMRES con reinicio, denominado GMRES(m). GMRES(m) es un método popular para la resolución de sistemas lineales $Ax = b$. Varios autores han propuesto el cambio de m y en general todos obtuvieron mejores resultados que usar un valor de m fijo. La propuesta de este trabajo consiste en tratar m como un parámetro de control que varía de acuerdo con una ley inspirada en un controlador PD (proporcional derivativo). Al GMRES(m) modificado con esta regla lo denominamos PD-GMRES.

PD-GMRES mostró ser mejor que GMRES(m) y otros métodos probados que varían m en los experimentos realizados, excepto en los Problemas 7 y 10 (ver la Figura 4.1). Aún cuando la variación de m no garantiza la convergencia, cambiar m es mucho más simple que usar LQRES o LQRESD (§B).

La estrategia de control PD es una regla simple que utiliza las normas de los tres residuos anteriores. El ajuste realizado por el controlador PD tiende a disminuir m a partir de un valor inicial. Si el valor calculado llega a un valor mínimo, el valor inicial es incrementado y la regla comienza a disminuir m a partir del nuevo valor inicial. Es importante resaltar que para el Problema 15 (ver 4.4.1), PD-GMRES es el único método que logra converger. Sin embargo, no resuelve el problema del estancamiento de forma general, como se puede ver en (4.4.2).

La justificación de que el cambio de m es mejor que tenerlo fijo, está en la observación hecha por [BJK09, GRLB08] de que cuando ocurre estancamiento se tienen residuos que si bien son decrecientes, son aproximadamente paralelos. De esta forma, cambiar la dimensión del subespacio de Krylov trata de evitar esa repetición de direcciones acelerando la convergencia.

5.1 Principales contribuciones

Las principales contribuciones de este trabajo son:

- la propuesta de PD-GMRES que utiliza una regla simple en términos de cálculo y memoria para cambiar el valor de m . Esta regla está inspirada en un controlador PD. La misma, aunque puede aumentar el valor de m , calcula las soluciones en un tiempo competitivo con respecto a las reglas comparadas. En la mayoría de los casos de prueba, PD-GMRES encuentra la solución en un tiempo menor.
- el modelado de $\text{GMRES}(m)$ como parte de un sistema de control cuyo residuo es llevado a cero por GMRES.

5.2 Trabajos futuros

Posibles trabajos futuros son:

- Aplicar la teoría de Control para variar parámetros de otros métodos con recurrencias largas.
- Tratar de aplicar técnicas de ajuste automático (automatic tuning) para la selección de parámetros de la regla propuesta.
- El estudio de convergencia del método. Si bien la aproximación de la derivada puede considerarse como una variación de una función de Lyapunov, cuando hay estancamiento su valor es cero o prácticamente cero, por lo que no se asegura convergencia. Por otra parte, la regla propuesta aumenta m , pero llevar m hasta la dimensión de la matriz del sistema no es computacionalmente práctico.
- Utilizar la parte integrativa en el controlador. Si bien el cociente de normas de residuos solo puede ser interpretado como el coseno del ángulo entre dos residuos consecutivos, la parte integral utiliza la historia de los residuos. Utilizar los valores de los cosenos con los ángulos anteriores permitiría saber si hay buena convergencia con respecto a los residuos anteriores. Sin embargo, su cálculo es costoso porque se necesita hacer un producto interior que es una operación vectorial.
- El estudio de la robustez cuando la operación de multiplicación matriz-vector es inexacta.

ANEXO

PD-GMRES. EXPERIMENTOS PARA SELECCIONAR α y γ

Incluimos aquí los experimentos realizados para seleccionar los parámetros α , γ de PD-GMRES. Esos parámetros son utilizados en la ecuación 3.14 de la sección 3.5.3, que calcula un nuevo valor para m a partir de su valor anterior. Para facilitar la lectura, la repetimos:

$$m_j = m_{j-1} + \left[\alpha \frac{\|r_{j-1}\|}{\|r_{j-2}\|} + \gamma \frac{\|r_{j-1}\| - \|r_{j-3}\|}{2\|r_{j-2}\|} \right] \quad (5.1)$$

Los valores probados fueron: $\alpha \in [2, 9]$ y $\gamma \in [-9, -2]$. Los valores para γ son negativos porque buscamos disminuir m y los valores de las normas de los residuos generados por GMRES(m) forman una sucesión de valores monotónicamente no crecientes.

Aunque la elección de esos intervalos fue arbitraria, se tuvo en cuenta:

- Parte proporcional: El Cociente $\frac{\|r_{j-1}\|}{\|r_{j-2}\|}$ está acotado por 1 dado que $\cos(\angle(r_{j-1}, r_{j-2})) = \frac{\|r_{j-1}\|}{\|r_{j-2}\|}$. Por otra parte, la parte proporcional es parecida a la regla de Baker sin necesidad de ejecutar las sentencias *if* y sin usar un valor fijo para m . Baker et. al. utilizan el valor de disminución 3 [BJK09] y ese valor está en el intervalo $[2, 9]$.
- Parte derivativa: No está acotada. En los experimentos realizados, la mayor reducción de la norma de los residuos es entre el residuo inicial r_0 y el residuo r_1 , luego la variación de las normas no es muy grande. Para que el efecto de la parte derivativa pueda notarse y que no sea extremadamente grande en comparación con la parte proporcional, elegimos valores enteros del intervalo $[-9, -2]$

Ejecutamos PD-GMRES con 64 pares (α, γ) para cada problema y medimos los tiempos de ejecución. También ejecutamos GMRES(30) para cada problema. Usamos el tiempo relativo que es el cociente entre $\frac{t_{PD-GMRES(m)}}{t_{GMRES(30)}}$. Buscamos el par de enteros (α, γ) que minimice la sumatoria de tiempos relativos de los problemas.

$$\min_{(\alpha \in [2,9], \gamma) \in [-9, -2]} \sum_{prob=1}^{15} \frac{t(PD - GMRES(m))}{t(GMRES(30))} \quad (5.2)$$

Este apéndice incluye un resumen de los tiempos relativos, los mejores 10 tiempos relativos, con sus respectivos valores de α y γ . Además por cada problema, se incluyen los tiempos de ejecución para cada par (α, γ) y los respectivos mejores 10 tiempos.

Todos los experimentos fueron hechos utilizando Matlab para Linux en una pc con Kubuntu 9.1 instalado. El procesador de la PC es Intel Centrino Core 2 Duo y su memoria RAM es de 2Mb.

5.3 Elección

El mejor par (α, γ) , según el criterio utilizado 5.2, es $(3, -9)$. Aunque la mayoría de las veces, la derivada tiene un valor pequeño, no está acotada. Un valor grande podría producir variaciones grandes en los valores de m y la regla que usa PD-GMRES podría hacer crecer rápidamente el valor de m . Tratamos hacer crecer m de una forma conservadora, por lo que buscamos utilizar un valor distinto de -9 que dé un peso un poco menor a la parte derivativa del controlador. Debido a eso, probamos el par $(3, -5)$ y comparamos sus resultados con los del par $(3, -9)$. Los tiempos obtenidos usando $(3, -5)$ no son muy diferentes que los tiempos obtenidos por el par $(3, -9)$ y en algunos casos son mejores. Entonces, elegimos el par $(3, -5)$ como el par (α, γ) de PD-GMRES(m). Este par obtiene buenos tiempos, pero hay que notar que para cada problema hay diferentes valores de (α, γ) que mejores tiempos.

Tabla 5.1: Tiempos por problema y algoritmo. $\alpha = 3$, $\gamma = -9$, 5 mediciones de tiempo.

Prob.	GMRES(30)		Baker gmres(m)		PD-GMRES(m)		Gonzalez gmres(m)	
	Tiempo (seg.)		Tiempo (seg.)		Tiempo (seg.)		Tiempo (seg.)	
	Prom.	Var.	Prom.	Var.	Prom.	Var.	Prom.	Var.
1	17.7437	0.2249	9.2157	0.1691	8.3891	0.0391	11.5592	0.6475
2	2.3194	0.2304	1.6069	0.0613	1.2020	0.0029	2.8601	0.0785
3	36.8237	4.1522	36.8000	0.8138	22.7708	0.7612	42.8250	1.8919
4	18.1224	2.1586	7.4236	0.6125	5.4094	0.1202	18.5770	2.6519
5	13.4254	3.8832	7.1122	0.0791	5.7252	0.0635	13.7273	7.7392
6	7.8283	0.2939	6.4592	0.1335	5.6803	0.1613	7.4054	0.7335
7	7.1557	0.1997	6.2200	0.0087	6.6989	0.2364	6.2554	0.0819
8	99.4634	44.9464	42.5339	11.1245	29.1733	1.7404	103.1257	137.2238
9	136.1778	3.6739	109.1609	28.4982	62.8267	22.7748	135.9549	3.5176
10	5.0067	0.1503	5.0617	0.0418	4.6538	0.1044	5.7373	0.2307
11	2.2404	0.0062	1.4729	0.0082	1.4016	0.0149	2.2317	0.0103
12	0.3702	0.0013	0.3600	0.0004	0.2411	0.0001	0.4044	0.0004
13	40.9925	82.4614	19.9110	0.6172	17.6929	0.3600	30.4704	6.7038
14	27.9854	1.0376	7.8887	0.1657	6.7007	0.3772	14.2184	1.0643
15	20.4976	0.0237	19.8914	0.0288	18.1970	6.9985	20.1603	0.0258

Tabla 5.2: Tiempos por problema y algoritmo. $\alpha = 3$, $\gamma = -5$, 5 mediciones de tiempo.

Prob.	GMRES(30)		Baker gmres(m)		PD-GMRES(m)		Gonzalez gmres(m)	
	Tiempo (seg.)		Tiempo (seg.)		Tiempo (seg.)		Tiempo (seg.)	
	Prom.	Var.	Prom.	Var.	Prom.	Var.	Prom.	Var.
1	18.1387	0.2225	9.1783	0.2854	9.0660	0.2560	11.8612	0.6479
2	2.3045	0.0252	1.7723	0.0183	1.2431	0.0118	2.6682	0.1520
3	39.2656	2.6792	35.9989	0.1007	25.8835	1.2464	42.4748	3.4607
4	18.0162	0.7852	7.4224	0.6425	5.6771	0.0692	20.0125	3.7830
5	15.1051	3.4572	7.2497	0.1061	5.8543	0.0157	13.6404	7.5766
6	8.1578	0.0775	6.3799	0.0324	5.6494	0.0713	7.0320	0.0387
7	7.2520	0.1056	6.2792	0.0114	6.8923	0.1572	6.0718	0.1167
8	94.2983	12.3607	43.2208	10.7898	26.3677	5.0020	95.2004	32.9560
9	140.4406	4.7902	113.0927	11.9849	69.4782	2.4265	140.2685	5.7715
10	5.0658	0.1379	5.0270	0.1072	4.5223	0.0366	5.5111	0.2848
11	2.2922	0.0028	1.3700	0.0156	1.3356	0.0087	2.2822	0.0342
12	0.3957	0.0010	0.3540	0.0002	0.2450	0.0005	0.4147	0.0005
13	37.9822	32.7382	19.9589	0.8582	17.2837	0.9935	36.0783	19.9427
14	28.5128	0.4900	8.1348	0.2925	7.4021	0.5123	14.1173	5.6670
15	20.4496	0.0042	19.9419	0.0539	18.1827	4.0652	20.0906	0.0091

Notar que en la tabla de tiempo, los problemas más grandes (matrices con mayor dimensión) tienen varianzas relativamente grandes. Esto se debería a que para hacer los cálculos el programa usaría memoria auxiliar (manejada directamente por el programa Matlab o por el sistema operativo) pero eso no está bajo el control del programador.

5.4 Resumen de tiempos relativos

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	9.9933	9.0770	8.9532	9.1054	9.1249	9.4848	9.2532	9.8774
	3	8.6881	9.5174	9.4579	9.2113	9.0489	9.1039	9.3484	9.2673
	4	8.9292	9.1588	9.3705	9.4879	9.4287	9.3195	9.1447	9.2452
	5	9.5002	9.1723	9.2949	9.2319	9.5033	9.6774	9.5098	9.5893
	6	9.3876	9.4667	9.5873	9.2314	9.2485	9.3048	9.6437	9.8889
	7	9.6774	9.4461	9.5015	9.7570	9.8140	9.7507	9.6046	9.3199
	8	9.5715	9.4566	9.9125	9.7014	9.9797	10.0522	9.8257	10.0500
	9	10.2221	9.8085	10.1873	10.3653	10.5971	11.3001	11.2451	11.2556

Tabla 5.3: Sumatoria de tiempos relativos. $tabla(\alpha, \gamma) = \sum_{prob=1}^{15} \frac{t(PD-GMRES)}{t(GMRES(30))}$

$tiempo(\alpha, \gamma)$	α	γ
8.6881	3	-9
8.9292	4	-9
8.9532	2	-7
9.0489	3	-5
9.0770	2	-8
9.1039	3	-4
9.1054	2	-6
9.1249	2	-5
9.1447	4	-3
9.1588	4	-8

Tabla 5.4: Menores tiempos relativos. $tiempo(\alpha, \gamma) = \sum_{prob=1}^{15} \frac{t(PD-GMRES)}{t(GMRES(30))}$

5.5 Tiempos de ejecución de GMRES(30)

Problema Nro	$t(GMRES(30))$
1	18.9662
2	1.7762
3	35.1057
4	17.4037
5	11.4426
6	9.0011
7	7.3444
8	93.4329
9	138.6711
10	5.1355
11	2.1969
12	0.3787
13	56.0413
14	27.8590
15	17.5404

Tabla 5.5: Tiempo de ejecución de $GMRES(30)$ en segundos

5.6 Problema 1. Matriz add20

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	9.1124	9.4739	8.6595	9.1090	9.4072	8.7009	9.1247	9.6358
	3	9.0287	8.3207	8.9940	8.9089	8.8462	9.6092	8.9637	9.1248
	4	8.9611	8.4491	8.0750	8.5609	8.7568	9.2072	9.0341	9.4289
	5	9.2635	9.3139	9.0249	9.5615	8.6650	9.2379	8.8206	9.2880
	6	8.8409	9.0628	9.0646	9.1710	8.9127	9.1201	9.6596	8.8432
	7	9.3630	9.0533	9.0861	9.3618	8.3633	9.1152	9.1042	8.6472
	8	9.1858	9.8540	9.2594	9.4584	9.4601	9.6143	8.9122	9.0457
	9	8.9150	9.1797	9.3092	9.5774	9.7818	9.5269	10.1061	9.9854

Tabla 5.6: add20.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
8.0750	4	-7
8.3207	3	-8
8.3633	7	-5
8.4491	4	-8
8.5609	4	-6
8.6472	7	-2
8.6595	2	-7
8.6650	5	-5
8.7009	2	-4
8.7568	4	-5

Tabla 5.7: add20.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.7 Problema 2. Matriz Cdde1

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	1.5863	1.2905	1.3506	1.3555	1.2837	1.2947	1.5042	1.5615
	3	1.2280	1.3615	1.2047	1.3133	1.4361	1.2863	1.4423	1.1961
	4	1.1479	1.2301	1.3578	1.3303	1.2969	1.3550	1.2334	1.1483
	5	1.2342	1.2091	1.1615	1.1990	1.2665	1.2603	1.2631	1.1516
	6	1.3051	1.1593	1.3810	1.2070	1.1891	1.0698	1.2754	1.5013
	7	1.0903	1.1567	1.0746	1.1439	1.2688	1.3149	1.1801	1.1363
	8	1.0625	1.0779	1.3630	1.0597	1.2690	1.2401	1.1221	1.2910
	9	1.3769	1.2653	1.4734	1.4670	1.4806	1.6780	1.5970	1.7637

Tabla 5.8: cdde1.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
1.0597	8	-6
1.0625	8	-9
1.0698	6	-4
1.0746	7	-7
1.0779	8	-8
1.0903	7	-9
1.1221	8	-3
1.1363	7	-2
1.1439	7	-6
1.1479	4	-9

Tabla 5.9: cdde1.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.8 Problema 3. Matriz Circuit_2

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	25.8248	24.3137	26.7275	27.0950	27.7192	29.9316	31.5299	37.4870
	3	24.0981	23.1399	25.2738	25.8074	25.5549	28.1099	29.0866	32.4266
	4	22.7782	24.3831	23.7876	24.4652	24.9476	25.9578	27.8487	30.3675
	5	22.2450	22.8903	23.0897	24.2370	25.7785	25.0809	27.8775	29.3623
	6	22.1081	22.8574	24.5182	23.5251	24.7356	24.5649	27.5704	27.3872
	7	21.9014	22.5205	24.1175	24.5486	24.2917	25.5954	26.7287	27.2581
	8	21.6161	22.4257	22.8447	24.9837	24.0273	25.1234	25.8160	27.4525
	9	22.7354	23.6294	24.2492	24.8057	26.1086	26.1223	26.5231	29.6933

Tabla 5.10: circuit_2.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
21.6161	8	-9
21.9014	7	-9
22.1081	6	-9
22.2450	5	-9
22.4257	8	-8
22.5205	7	-8
22.7354	9	-9
22.7782	4	-9
22.8447	8	-7
22.8574	6	-8

Tabla 5.11: circuit_2.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.9 Problema 4. Matriz Fpga_trans_01

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	7.2501	6.0193	6.9016	6.4935	7.1315	6.3855	6.8704	6.6769
	3	6.1711	5.6154	5.7403	6.5627	6.7677	5.7800	6.0342	6.9109
	4	5.7198	5.5928	5.9306	5.8678	6.0982	5.8737	6.0195	5.9499
	5	5.4572	5.2796	5.7017	5.3091	5.6335	5.4257	5.7897	5.9110
	6	5.2667	5.6699	5.7304	5.6479	5.5371	5.8546	5.1423	5.3740
	7	5.3424	5.4568	5.1945	5.2849	5.4097	5.3056	5.3588	5.7562
	8	5.2590	5.3472	5.4917	5.1778	5.4960	5.7698	5.3755	5.7209
	9	5.4860	5.4503	5.6264	6.1171	5.2417	5.5629	5.6365	5.7455

Tabla 5.12: fpga_trans_01.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
5.1423	6	-3
5.1778	8	-6
5.1945	7	-7
5.2417	9	-5
5.2590	8	-9
5.2667	6	-9
5.2796	5	-8
5.2849	7	-6
5.3056	7	-4
5.3091	5	-6

Tabla 5.13: fpga_trans_01.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.10 Problema 5. Matriz Orsirr_1

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	7.5900	6.7778	6.2254	6.5786	6.5237	6.3487	7.0570	6.4537
	3	5.7475	5.8847	5.8858	6.1307	5.8146	6.1502	5.8437	5.9368
	4	5.9986	5.8616	5.5474	5.7954	5.8888	6.2224	5.8000	5.8416
	5	5.5092	5.7178	6.2862	5.7611	5.5570	6.3667	5.5527	5.8448
	6	5.4935	5.4959	6.0335	5.5768	5.7032	6.0392	5.7159	6.0614
	7	6.0024	5.8011	5.7813	5.9016	6.1672	5.7985	5.8781	5.7792
	8	5.5960	5.7549	5.9938	5.6924	5.9040	5.7997	5.8252	5.9250
	9	5.8514	5.8476	6.3067	6.4844	6.2980	6.2878	6.7368	6.3466

Tabla 5.14: orsirr_1.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
5.4935	6	-9
5.4959	6	-8
5.5092	5	-9
5.5474	4	-7
5.5527	5	-3
5.5570	5	-5
5.5768	6	-6
5.5960	8	-9
5.6924	8	-6
5.7032	6	-5

Tabla 5.15: orsirr_1.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.11 Problema 6. Matriz Orsreg_1

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	5.7701	6.1957	5.9881	5.9584	5.8789	6.4123	6.4906	6.4755
	3	5.8037	5.9632	5.6947	6.1101	6.1057	5.7968	5.9565	6.0332
	4	5.9935	5.9465	5.9414	6.1115	6.2983	5.9472	6.1524	6.4233
	5	6.3226	5.8710	5.9426	5.6172	6.1887	6.1465	6.2853	5.9779
	6	5.7907	5.8552	6.0794	5.8225	6.2597	5.8980	6.0747	6.0845
	7	6.1782	5.9512	5.9620	5.8173	6.4259	5.8608	6.0551	6.0279
	8	6.2925	6.1551	6.3019	5.9306	6.1912	5.9250	6.2333	6.2088
	9	6.3514	6.6521	6.7920	6.3988	6.3484	6.6160	6.3446	6.4114

Tabla 5.16: orsreg_1.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
5.6172	5	-6
5.6947	3	-7
5.7701	2	-9
5.7907	6	-9
5.7968	3	-4
5.8037	3	-9
5.8173	7	-6
5.8225	6	-6
5.8552	6	-8
5.8608	7	-4

Tabla 5.17: orsreg_1.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.12 Problema 7. Matriz Pde2961

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	8.6541	7.2618	7.4163	6.7139	6.9095	7.4818	8.0446	7.1475
	3	6.8829	7.0540	7.2222	7.2289	6.6265	6.8068	6.9876	7.3310
	4	7.0306	6.9045	7.3111	7.1158	7.2306	6.6786	6.7610	6.7145
	5	7.3922	7.4830	7.5408	7.7853	7.0869	7.2274	7.2713	6.7152
	6	7.6136	7.5416	7.5045	7.0576	7.0143	7.0827	7.2858	7.0709
	7	7.3579	7.1738	7.1347	7.2897	7.2120	7.8800	7.2937	7.0556
	8	6.9915	6.9022	7.2564	7.1936	7.0266	7.3431	7.4094	8.1576
	9	8.3670	7.3506	7.2293	6.9155	7.2009	7.5219	8.0111	8.1661

Tabla 5.18: pde2961.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
6.6265	3	-5
6.6786	4	-4
6.7139	2	-6
6.7145	4	-2
6.7152	5	-2
6.7610	4	-3
6.8068	3	-4
6.8829	3	-9
6.9022	8	-8
6.9045	4	-8

Tabla 5.19: pde2961.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.13 Problema 8. Matriz Raefsky1

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	32.9456	32.0912	26.1548	23.1743	24.0818	25.8203	25.3873	30.3965
	3	27.5274	29.1414	29.1394	30.2545	30.4138	27.7528	27.6350	28.1138
	4	37.7081	35.5093	32.7549	36.8632	35.4236	37.6494	30.2080	35.0229
	5	38.6072	33.9807	37.6855	32.4355	30.9802	40.0465	39.7440	35.1918
	6	33.5572	38.8508	34.1816	37.3028	38.2261	35.7473	35.6457	28.1017
	7	38.6837	35.3903	39.0740	47.8397	34.9564	37.3400	33.6186	38.3690
	8	40.2391	38.3444	43.7099	43.2453	33.1200	41.7459	37.7707	40.6119
	9	43.2851	44.2043	41.8408	41.9384	37.6187	50.3931	41.9500	42.8111

Tabla 5.20: raefsky1.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
23.1743	2	-6
24.0818	2	-5
25.3873	2	-3
25.8203	2	-4
26.1548	2	-7
27.5274	3	-9
27.6350	3	-3
27.7528	3	-4
28.1017	6	-2
28.1138	3	-2

Tabla 5.21: raefsky1.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.14 Problema 9. Matriz Raefsky2

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	50.3869	51.1558	42.9663	48.2038	49.0124	50.6449	51.9158	62.8643
	3	62.9695	65.1750	69.3956	70.2214	62.4616	72.5152	74.1929	61.3672
	4	71.8042	66.6983	80.0821	70.8089	79.1926	78.8578	79.5169	72.8385
	5	74.9215	72.6534	72.2906	70.3261	84.7337	81.1971	78.8377	81.2869
	6	67.7373	68.8593	79.3767	73.0470	74.0935	75.0127	74.8093	80.5275
	7	75.1249	79.3338	72.3069	72.0640	78.4651	78.6297	68.0960	76.0215
	8	70.4985	67.5677	72.1533	76.2182	71.1474	75.6824	76.8120	72.6755
	9	67.8661	68.9237	67.7228	77.6144	76.5451	81.7872	83.8732	76.8884

Tabla 5.22: raefsky2.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
42.9663	2	-7
48.2038	2	-6
49.0124	2	-5
50.3869	2	-9
50.6449	2	-4
51.1558	2	-8
51.9158	2	-3
61.3672	3	-2
62.4616	3	-5
62.8643	2	-2

Tabla 5.23: raefsky2.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.15 Problema 10. Matriz Rdb2048

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	5.8441	4.7413	4.4591	4.7989	5.1221	4.5066	4.5539	5.2811
	3	4.5698	4.7685	4.5862	4.7721	4.8062	4.7730	4.8014	5.2199
	4	4.6942	4.7730	4.4647	4.8726	5.0090	4.4657	4.3913	4.9189
	5	4.7520	4.7426	4.8304	4.9487	5.0007	5.0803	4.6826	5.0850
	6	5.2024	4.6802	4.7480	5.2213	4.8693	4.3269	4.5456	4.5702
	7	4.7482	4.4438	4.5584	4.6257	4.7824	5.0053	4.8288	4.6342
	8	4.9605	4.9102	4.6916	4.6234	5.1893	5.0358	4.7406	4.6619
	9	4.6183	4.6109	5.2683	4.8516	5.6438	5.1092	5.4323	5.1127

Tabla 5.24: rdb2048.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
4.3269	6	-4
4.3913	4	-3
4.4438	7	-8
4.4591	2	-7
4.4647	4	-7
4.4657	4	-4
4.5066	2	-4
4.5456	6	-3
4.5539	2	-3
4.5584	7	-7

Tabla 5.25: rdb2048.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.16 Problema 11. Matriz Sherman4

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	1.9042	1.4091	1.4820	1.5852	1.2901	1.4705	1.2987	1.4246
	3	1.6404	1.6063	1.2637	1.3401	1.4407	1.2878	1.3001	1.4497
	4	1.5493	1.3550	1.5178	1.5402	1.4707	1.2816	1.5564	1.5219
	5	1.5441	1.3829	1.3610	1.5622	1.6986	1.5520	1.4380	1.6481
	6	1.5636	1.7274	1.5507	1.3891	1.3470	1.4551	1.4641	1.5533
	7	1.6559	1.7081	1.5597	1.6041	1.7878	1.5535	1.5307	1.3038
	8	1.6474	1.5780	1.7399	1.5238	1.8635	1.6514	1.4727	1.3609
	9	1.8177	1.7087	1.9262	1.7085	1.8569	2.0654	1.9259	1.8590

Tabla 5.26: sherman4.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
1.2637	3	-7
1.2816	4	-4
1.2878	3	-4
1.2901	2	-5
1.2987	2	-3
1.3001	3	-3
1.3038	7	-2
1.3401	3	-6
1.3470	6	-5
1.3550	4	-8

Tabla 5.27: sherman4.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.17 Problema 12. Matriz Steam2

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	0.3137	0.2964	0.2831	0.2970	0.2934	0.4106	0.2926	0.3289
	3	0.2832	0.4352	0.4405	0.2871	0.2820	0.3206	0.2626	0.2629
	4	0.2469	0.2648	0.3163	0.2988	0.2858	0.2526	0.2558	0.2936
	5	0.2452	0.2476	0.2489	0.2455	0.2536	0.2522	0.2597	0.2676
	6	0.2458	0.2946	0.2911	0.2506	0.2554	0.2585	0.2695	0.3637
	7	0.2537	0.2259	0.2345	0.3012	0.2823	0.2622	0.3071	0.2456
	8	0.2730	0.2311	0.2663	0.3483	0.3522	0.3420	0.3061	0.3395
	9	0.3036	0.3038	0.2620	0.3065	0.3962	0.4983	0.4081	0.4244

Tabla 5.28: steam2.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
0.2259	7	-8
0.2311	8	-8
0.2345	7	-7
0.2452	5	-9
0.2455	5	-6
0.2456	7	-2
0.2458	6	-9
0.2469	4	-9
0.2476	5	-8
0.2489	5	-7

Tabla 5.29: steam2.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.18 Problema 13. Matriz Wang2

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	15.5074	14.7866	16.5890	16.6881	18.1382	15.1031	14.1036	16.2754
	3	15.1154	16.8642	17.3246	16.9363	17.4406	17.0763	19.7105	15.1361
	4	18.1388	16.7190	16.2620	18.8859	19.9097	17.5416	16.2559	14.2218
	5	20.3978	19.6427	19.1042	19.0401	19.9920	18.3791	21.1071	19.2953
	6	20.3268	20.8336	20.3509	20.8890	20.0179	19.3293	19.3692	18.6492
	7	21.4085	21.7742	19.9058	19.1009	19.6360	20.1638	19.8924	21.1160
	8	21.1576	20.3097	21.2009	20.2932	19.2466	22.2853	19.4327	18.7292
	9	23.2235	21.0665	22.0154	20.0734	21.8549	21.0831	22.0650	21.2255

Tabla 5.30: wang2.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
14.1036	2	-3
14.2218	4	-2
14.7866	2	-8
15.1031	2	-4
15.1154	3	-9
15.1361	3	-2
15.5074	2	-9
16.2559	4	-3
16.2620	4	-7
16.2754	2	-2

Tabla 5.31: wang2.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.19 Problema 14. Matriz Watt_1

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	9.4119	8.1320	8.4404	6.4725	6.0980	6.7824	6.8461	6.2582
	3	7.5139	8.0133	9.3596	7.9299	7.1389	7.6409	7.7698	8.1966
	4	7.9733	7.7628	7.4875	8.4023	8.1297	8.7756	7.3946	7.3732
	5	8.5721	7.6327	7.9199	7.4562	8.4631	8.3224	6.4946	7.9187
	6	7.5520	7.8801	6.6099	7.0447	8.4796	7.9440	9.2138	8.0666
	7	9.2867	8.7058	9.9589	8.3824	9.5535	8.6707	8.7526	8.1668
	8	8.2961	8.7548	8.6237	8.8259	8.9350	8.3708	8.5099	9.1005
	9	8.4501	8.5618	9.4152	9.1921	8.7764	9.3432	10.2627	9.7716

Tabla 5.32: watt_1.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
6.0980	2	-5
6.2582	2	-2
6.4725	2	-6
6.4946	5	-3
6.6099	6	-7
6.7824	2	-4
6.8461	2	-3
7.0447	6	-6
7.1389	3	-5
7.3732	4	-2

Tabla 5.33: watt_1.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

5.20 Problema 15. Matriz Young3c

		γ							
		-9	-8	-7	-6	-5	-4	-3	-2
α	2	14.4051	16.2694	15.4388	17.5393	18.0598	17.6453	14.1202	16.5457
	3	12.6826	17.7829	18.1792	17.1299	15.5883	15.2744	18.9976	16.2509
	4	15.4895	21.1971	19.9801	19.5818	17.1939	20.3618	18.1106	15.9977
	5	22.4269	20.2666	20.6353	19.0308	19.6038	21.5513	20.6551	20.1316
	6	21.7327	20.3326	20.0330	19.5782	19.2360	22.7501	22.9536	22.0829
	7	24.6792	23.2625	24.9742	23.8934	22.7567	21.9808	21.8091	21.1947
	8	24.2053	24.5318	24.4477	22.0333	22.5421	22.9584	25.5620	24.0994
	9	25.5897	21.8154	22.8238	26.8142	23.0934	24.3666	26.6859	25.9052

Tabla 5.34: young3c.mat, PD-GMRES, $tiempo(\alpha, \gamma)$ en segundos

$tiempo(\alpha, \gamma)$	α	γ
12.6826	3	-9
14.1202	2	-3
14.4051	2	-9
15.2744	3	-4
15.4388	2	-7
15.4895	4	-9
15.5883	3	-5
15.9977	4	-2
16.2509	3	-2
16.2694	2	-8

Tabla 5.35: young3c.mat, 10 menores tiempos, $t(\alpha, \gamma)$ en segundos

REFERENCIAS BIBLIOGRÁFICAS

- [and97] L. N. Trefethen and. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997.
- [BBG⁺09] Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc Web page, 2009. <http://www.mcs.anl.gov/petsc>.
- [Ber05] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Massachusetts, 2005.
- [BJK09] A. H. Baker, E. R. Jessup, and Tz. V. Kolev. A simple strategy for varying the restart parameter in gmres(m). *J. Comput. Appl. Math.*, 230(2):751–761, 2009.
- [BJM05] A. H. Baker, E. R. Jessup, and T. Manteuffel. A technique for accelerating the convergence of restarted gmres. *SIAM J. Matrix Anal. Appl.*, 26(4):962–984, 2005.
- [BK04] A. Bhaya and E. Kaszkurewicz. A control theoretic approach to the design of zero finding numerical methods. *IEEE Transactions on Automatic Control*, 52(6):1014–1026, 2004.
- [Bre98] C. Brezinski. Hybrid methods for solving systems of equations. In G. Winter Althaus and E. Spedicato, editors, *Algorithms for Sparse Large Scale Linear Algebraic Systems*, pages 271–290. Kluwer, Dordrecht, 1998.
- [bTDfUoFfAR] Maintained by T. Davis from Univ. of Florida and Y. Hu from ATT Research. The university of florida matrix collection. Sitio web <http://www.cise.ufl.edu/research/sparse/matrices/>.
- [CB04] M. T. Vespucci C.G. Broyden. In Charles George Broyden and Maria Teresa Vespucci, editors, *Krylov Solvers for Linear Algebraic Systems - Krylov Solvers*, volume 11 of *Studies in Computational Mathematics*. Elsevier, 2004.

- [Dem97] J. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997.
- [EES00] Michael Eiermann, Oliver G. Ernst, and Olaf Schneider. Analysis of acceleration strategies for restarted minimal residual methods. *Journal of Computational and Applied Mathematics*, 123(1-2):261 – 292, 2000.
- [Emb03] Mark Embree. The tortoise and the hare restart gmres. *SIAM Review*, 45(2):259–266, 2003.
- [GC05] T.T. Gonzalez and R. D. Da Cunha. Seleção dinâmica da dimensão do subespaço de krylov no método gmres(m) e suas variantes. *TEMA Tend. Mat. Apl. Comput.*, 26(4):962–984, 2005.
- [Gre97] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, 1997.
- [GRLB08] M.A. Gomes-Ruggiero, V.L. Rocha Lopes, and J. V. Toledo Benavides. A safeguard approach to detect stagnation of gmres(m) with applications in newton-krylov methods. *Computational and Applied Mathematics*, 27(2):175–179, 2008.
- [HJ10] Uwe Helmke and Jens Jordan. Control and stabilization of linear equation solvers. In Jan Willems, Shinji Hara, Yoshito Ohta, and Hisaya Fujioka, editors, *Perspectives in Mathematical System Theory, Control, and Signal Processing*, volume 398 of *Lecture Notes in Control and Information Sciences*, pages 73–82. Springer Berlin / Heidelberg, 2010.
- [HJL06] U. Helmke, J. Jordan, and A. Lanzon. A control theory approach to linear equation solvers. In *Proc. of 17th International Symposium on Mathematical Theory of Networks and Systems-Kyoto Japan*, pages 1401–1407, 2006.
- [HN03] M. Habu and T. Nodera. Gmres(m) with changing restart cycle adaptively. In *Proceedings of Allgoritmy 2000. Conference on Scientific Computing.*, pages 1105–1113. Poland, 9 2003.
- [IM98] Ilse C. F. Ipsen and Carl D. Meyer. The idea behind krylov methods. *American Mathematical Monthly*, pages 889–899, 1998.
- [MfNU] Statistics Math and Computational Science from NIST. USA. Matrix market. Sitio web <http://math.nist.gov/MatrixMarket/>.

- [MN00] K. Moriya and T. Nodera. New adaptive gmres(m) method with choosing suitable restart cycle m. In *Parallel Processing and Applied Mathematics 2003 Proceedings*, pages 254–263. Slovak Univ.of Tech, Bratislava, Comenius Univ., Bratislava, and Univ. of Technology, Zvolen., Vysoke Tatry-Podbanske, Slovakia, 9 2000.
- [Oga87] K. Ogata. *Dinámica de Sistemas*. Prentice Hall Hispanoamericana, Mexico, 1987.
- [Oga95] Katsuhiko Ogata. *Discrete Time Control Systems*. Prentice Hall International, Inc., New Jersey. USA, 1995.
- [PRS06] C.C. Paige, M. Rozloznik, and Z. Strakos. Modified gram-schmidt (mgs), least squares and backward stability of mgs-gmres. *SIAM J. Matrix Anal. and Appl.*, 28(1):264–284, 2006.
- [Saa03] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2003.
- [SS86] Yousef Saad and Martin H. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. on Sci. and Stat. Comp.*, 7(3):856–869, 1986.
- [SS03] Valeria Simoncini and Daniel B. Szyld. Theory of inexact krylov subspace methods and applications to scientific computing. *SIAM Journal on Scientific Computing*, 25(2):454–477, 2003.
- [SS07] Valeria Simoncini and Daniel B. Szyld. Recent computational developments in krylov subspace methods for linear systems. *Numer. Linear Algebra Appl*, 14:1–59, 2007.
- [SWK98] Maria Sosonkina, Layne T. Watson, and Rakesh K. Kapania. A new adaptive gmres algorithm for achieving high accuracy. *Numer. Linear Algebra Appl*, 5:275–297, 1998.
- [TB97] L.N. Trefethen and D. Bau. *Numerical linear algebra*. Society for Industrial Mathematics, 1997.
- [vdES04] Jasper van den Eshof and Gerard L. G. Sleijpen. Inexact krylov subspace methods for linear systems. *SIAM. J. Matrix Anal. & Appl.*, 26(1):125–153, 2004.

- [Vis06] A. Visioli. *Practical PID Control (Advances in IndustrialControl)*. Springer-Verlag, London, 2006.
- [VL10] E. Vecharynski and J. Langou. The cycle-convergence of restarted gmres for normal matrices is sublinear. *SIAM Journal on Scientific Computing*, 32(1):186–196, 2010.
- [Zit08] Jan Zitko. Some remarks on the restarted and augmented gmres method. *Electronic Transactions on Numerical Analysis.*, 31:221–227, 2008.
- [ZN05] Linjie Zhang and Takashi Nodera. A new adaptive restart for gmres(m) method. In Rob May and A. J. Roberts, editors, *Proc. of 12th Computational Techniques and Applications Conference CTAC-2004*, volume 46, pages C409–C425, may 2005. <http://anziamj.austms.org.au/V46/CTAC2004/Zhan> [May 24, 2005].

APENDICE A

RESIDUOS PARALELOS Y MATRICES SIMÉTRICAS

Teoremas que muestran que en el límite los residuos intercalados r_{j-1} y r_{j+1} son paralelos si la matriz A es simétrica [BJK09].

Lema 5 (Restricciones equivalentes). *Dada $A \in R^{n \times n}$ simétrica o anti-simétrica y $w, y \in R^n$, que se cumpla $w \perp AK_m(A, y)$ es equivalente a $y \perp A^T K_m(A^T, w)$.*

Demostración. Reescribiendo el subespacio de Krylov $AK_m(A, y) = \sum_{k=1}^m \alpha_k A^k y$ y usando el producto interior $\langle w, y \rangle = y^T w$, $w \perp AK_m(A, y)$ puede escribirse como

$$\begin{aligned}\langle w, \sum_{k=1}^m \alpha_k A^k y \rangle &= 0 \\ \sum_{k=1}^m \langle w, \alpha_k A^k y \rangle &= 0 \\ \sum_{k=1}^m \alpha_k y^T A^T w &= 0\end{aligned}\tag{A.1}$$

Reagrupando los términos en la expresión anterior

$$\begin{aligned}\sum_{k=1}^m \alpha_k y^T (A^T w) &= 0 \\ \sum_{k=1}^m \alpha_k \langle A^T w, y \rangle &= 0 \\ \langle y, \sum_{k=1}^m \alpha_k A^T w \rangle &= 0\end{aligned}\tag{A.2}$$

De la última ecuación obtenemos que $w \perp AK_m(A, y)$ es equivalente a $y \perp A^T K_m(A^T, w)$. □

Un ángulo secuencial $\angle(r_i, r_{i-1})$ es el ángulo entre dos residuos consecutivos r_i and r_{i-1} . Ángulos secuenciales pequeños indican estancamiento.

Un ángulo intercalado $\angle(r_{i+1}, r_{i-1})$ es el ángulo entre residuos intercalados, por ejemplo entre r_{i+1} y r_{i-1} . Si el ángulo intercalado es pequeño, significa que se están generando residuos casi paralelos.

Teorema 1 (Ángulo entre residuos secuenciales [BJK09]). *Sea r_{i+1} el residuo de GMRES(m) en la iteración $i + 1$, y r_i el residuo en la iteración i .*

Entonces el ángulo entre los residuos está dado por

$$\cos(\angle(r_{i+1}, r_i)) = \frac{\|r_{i+1}\|}{\|r_i\|}$$

Demostración. El residuo $i + 1$ de GMRES(m) puede ser escrito como $r_{i+1} = r_i - AV_m y$ donde $r_{i+1} \perp AV_m y$ porque y se obtiene usando mínimos cuadrados. $\min \|b - Ax\|$ es equivalente a $\min \|\cdot\|$

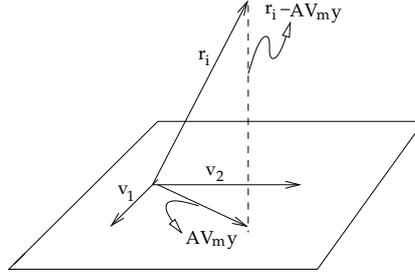


Figura A.1: $r_i - AV_m y \perp AV_m y$, where $r_{i+1} = r_i - AV_m y$

Luego calculamos

$$\cos(\angle(r_{i+1}, r_i)) = \frac{\langle r_{i+1}, r_i \rangle}{\|r_{i+1}\| \|r_i\|} \quad (\text{A.3})$$

Primero consideramos el producto interior

$$\begin{aligned} \langle r_{i+1}, r_i \rangle &= \langle r_{i+1}, r_{i+1} + AV_m y \rangle \\ \langle r_{i+1}, r_i \rangle &= \langle r_{i+1}, r_{i+1} \rangle + \langle r_{i+1}, AV_m y \rangle \\ \langle r_{i+1}, r_i \rangle &= \|r_{i+1}\|^2 - \langle r_{i+1}, AV_m y \rangle \end{aligned} \quad (\text{A.4})$$

Dado que $r_{i+1} \perp AV_m y$ $\langle r_{i+1}, AV_m y \rangle = 0$, la expresión para el producto interior es

$$\langle r_{i+1}, r_i \rangle = \|r_{i+1}\|^2 \quad (\text{A.5})$$

Llevando esto a la expresión del coseno, obtenemos

$$\begin{aligned} \cos(\angle(r_{i+1}, r_i)) &= \frac{\|r_{i+1}\|^2}{\|r_{i+1}\| \|r_i\|} \\ \cos(\angle(r_{i+1}, r_i)) &= \frac{\|r_{i+1}\|}{\|r_i\|} \end{aligned} \quad (\text{A.6})$$

□

Teorema 2 (Residuos intercalados paralelos cuando m_i es no-creciente[BJK09]).
Dado $A \in R^{n \times n}$ simétrica o antisimétrica, y m_i es no creciente para todas las ite-

raciones i , $Gmres(m)$ produce una secuencia de residuos r_i al final de ca iteración i tal que los ángulos intercalados tienden a cero, i.e.

$$\lim_{i \rightarrow \infty} \cos \phi_i = 0$$

Demostración. Podemos escribir

$$r_i = r_{i-1} - \sum_{k=1}^{m_i} \alpha_{i,k} A^k r_{i-1} \quad (\text{A.7})$$

Luego calculamos el producto interior $\langle r_{i+1}, r_{i+1} \rangle$ usando A.7

$$\begin{aligned} \langle r_{i+1}, r_{i+1} \rangle &= \langle r_{i+1}, r_i - \sum_{k=1}^{m_i} \alpha_{i,k} A^k r_i \rangle \\ \langle r_{i+1}, r_{i+1} \rangle &= \langle r_{i+1}, r_i \rangle - \langle r_{i+1}, \sum_{k=1}^{m_i} \alpha_{i,k} A^k r_i \rangle \end{aligned} \quad (\text{A.8})$$

$Gmres$ utiliza mínimos cuadrados para obtener el siguiente residuo, entonces $r_{i+1} \perp \sum_{k=1}^{m_i} \alpha_{i,k} A^k r_i$. Llevando esto a A.8

$$\langle r_{i+1}, r_{i+1} \rangle = \langle r_{i+1}, r_i \rangle \quad (\text{A.9})$$

Calculamos $\langle r_{i+1}, r_{i-1} \rangle$

$$\begin{aligned} \langle r_{i+1}, r_{i-1} \rangle &= \langle r_i - \sum_{k=1}^{m_{i+1}} \alpha_{i,k} A^k r_i, r_{i-1} \rangle \\ \langle r_{i+1}, r_{i-1} \rangle &= \langle r_i, r_{i-1} \rangle - \langle \sum_{k=1}^{m_{i+1}} \alpha_{i,k} A^k r_i, r_{i-1} \rangle \end{aligned} \quad (\text{A.10})$$

Dado que r_i también fue generado por $Gmres$, puede ser escrito como $r_i = r_{i-1} - \sum_{k=1}^{m_{i-1}} \alpha_{i,k} A^k r_{i-1}$, donde $r_i \perp \sum_{k=1}^{m_{i-1}} \alpha_{i,k} A^k r_{i-1}$. Usando el lema 5 y suponiendo que A es simétrica $r_{i-1} \perp \sum_{k=1}^{m_i} \alpha_{i,k} A^k r_i$. Aplicamos esto a A.10 para obtener

$$\begin{aligned} \langle r_{i+1}, r_{i-1} \rangle &= \langle r_i, r_{i-1} \rangle - \langle \sum_{k=1}^{m_{i+1}} \alpha_{i,k} A^k r_i, r_{i-1} \rangle \\ \langle r_{i+1}, r_{i-1} \rangle &= \langle r_i, r_{i-1} \rangle - \langle \sum_{k=1}^{m_i} \alpha_{i,k} A^k r_i, r_{i-1} \rangle - \langle \sum_{k=m_i+1}^{m_{i+1}} \alpha_{i,k} A^k r_i, r_{i-1} \rangle \\ \langle r_{i+1}, r_{i-1} \rangle &= \langle r_i, r_{i-1} \rangle - \langle \sum_{k=m_i+1}^{m_{i+1}} \alpha_{i,k} A^k r_i, r_{i-1} \rangle \end{aligned} \quad (\text{A.11})$$

Suponiendo que la secuencia m_i es no-creciente, $m_{i+1} \leq m_i$, la sumatoria en A.11 tiene cero términos. Por lo que tenemos

$$\langle r_{i+1}, r_{i-1} \rangle = \langle r_i, r_{i-1} \rangle \quad (\text{A.12})$$

Aplicando A.9 a $\langle r_i, r_{i-1} \rangle$, obtenemos $\langle r_i, r_{i-1} \rangle = \langle r_i, r_i \rangle$. Llevando la última

expresión a A.12

$$\langle r_{i+1}, r_{i-1} \rangle = \langle r_i, r_i \rangle \quad (\text{A.13})$$

El ángulo entre r_{i+1} y r_{i-1} se calcula usando

$$\cos(\angle(r_{i+1}, r_{i-1})) = \frac{\langle r_{i+1}, r_{i-1} \rangle}{\|r_{i+1}\| \|r_{i-1}\|} \quad (\text{A.14})$$

Reemplazando A.13 en A.14

$$\begin{aligned} \cos(\angle(r_{i+1}, r_{i-1})) &= \frac{\|r_i\|^2}{\|r_{i+1}\| \|r_{i-1}\|} \\ \cos(\angle(r_{i+1}, r_{i-1})) &= \frac{\frac{\|r_i\|}{\|r_{i+1}\|}}{\frac{\|r_{i-1}\|}{\|r_i\|}} \end{aligned} \quad (\text{A.15})$$

Sea $a_i = \frac{r_{i+1}}{r_i}$. Reemplazando en A.15

$$\cos(\angle(r_{i+1}, r_{i-1})) = \frac{a_{i-1}}{a_i} \quad (\text{A.16})$$

Como $\cos(\angle(r_{i+1}, r_{i-1})) \leq 1$ y usando A.16, tenemos $a_{i-1} \leq a_i$. Por el teorema 1 $\|r_{i+1}\| \leq \|r_i\|$, la secuencia a_i está acotada. Por lo tanto existe β tal que

$$\cos(\angle(r_{i+1}, r_{i-1})) = \lim_{i \rightarrow \infty} \frac{\beta}{\beta} = 1 \quad (\text{A.17})$$

Entonces, para un parámetro de reinicio fijo y A simétrica, los ángulos intercalados tienden a cero.

□

APENDICE B

LQRES y LQRES(D)(M)

Este apéndice incluye experimentos numéricos con los algoritmos LQRES y LQRES(D)(M) que son métodos inspirados en control, específicamente en control lineal cuadrático, que minimizan la norma euclidiana del residuo. Incluimos la deducción del control óptimo para un sistema lineal con costo cuadrático. También mencionamos sus condiciones de estabilidad. Luego pasamos a los algoritmos LQRES y LQRES(D)(M), incluimos los algoritmos y experimentos numéricos. Primero reproducimos el comportamiento de LQRES y LQRES(D)(M) con una matriz pequeña presentada originalmente por M. Embreé [Emb03] como un caso en el que GMRES(M) no converge. Luego mostramos ejemplos con matrices de problemas reales en los que GMRES(M) no converge. Finalmente presentamos observaciones sobre LQRES y LQRES(D).

B.1 Control Óptimo de Sistemas Lineales con costo cuadrático

Esta sección corresponde al libro [Oga95]. Si la matriz A tiene elementos complejos usamos la notación $A^H = A^* = \bar{A}$ para indicar la matriz hermitiana o traspuesta conjugada de A . Si la matriz A es real, $A^H = A^T$.

El problema de control óptimo cuadrático puede ser formulado como:

Dado el sistema lineal controlable en el espacio de estados

$$x_{k+1} = Gx_k + Hu_k, \quad x_0 = c \tag{B.1}$$

donde x_k es el vector de estado con dimensión $n \times 1$, u_k es el vector de control con dimensión $r \times 1$, G es una matriz no singular de dimensión $n \times n$ y H una matriz de dimensión $n \times r$,

encontrar la secuencia de control u_0, u_1, \dots, u_{N-1} , que minimiza la función de costo

$$J = \frac{1}{2}x_N^* S x_N + \frac{1}{2} \sum_{k=0}^{N-1} [x_k^* Q x_k + u_k^* R u_k] \tag{B.2}$$

donde Q es una matriz Hermitiana (o real simétrica) definida positiva o semidefinida positiva de dimensión $n \times n$, R es una matriz Hermitiana (o real simétrica) definida positiva de dimensión $r \times r$, S es una matriz Hermitiana (o real simétrica) definida positiva o semidefinida positiva de dimensión $n \times n$.

Las matrices Q , R y S miden la importancia de los vectores de estado x_k , de los vectores de control u_k ($k = 0, 1, \dots, N-1$) y el vector final x_N respectivamente. El estado inicial $x_0 = c$ es un estado arbitrario. Si el estado final x_N es fijo, se elimina de la ecuación B.2. Si el estado final queda en la ecuación B.2, indica que se busca un valor final lo más cercano posible a cero.

Para resolver el problema utilizando multiplicadores de Lagrange, el problema se formula como:

$$\min J = \frac{1}{2}x_N^*Sx_N + \frac{1}{2}\sum_{k=0}^{N-1}[x_k^*Qx_k + u_k^*Ru_k] \quad (\text{B.3})$$

sujeto a

$$x_{k+1} = Gx_k + Hu_k, \quad x_0 = c \quad (\text{B.4})$$

Usando multiplicadores de Lagrange, definimos la nueva función a minimizar:

$$L = \frac{1}{2}x_N^*Sx_N + \frac{1}{2}\sum_{k=0}^{N-1}\{[x_k^*Qx_k + u_k^*Ru_k] + \lambda_{k+1}^*[x_k^*Qx_k + u_k^*Ru_k] + [x_k^*Qx_k + u_k^*Ru_k]^*\lambda_{k+1}\} \quad (\text{B.5})$$

Para minimizar L hay que diferenciar con respecto a cada uno de los componentes de los vectores x_k , u_k y λ_k , pero es más conveniente y a la vez equivalente diferenciar con respecto a los componentes complejos conjugados. Por lo que hacemos:

$$\begin{aligned} \frac{\partial L}{\partial \bar{x}_{i,k}} & \quad i = 1, 2, \dots, n; \quad k = 1, 2, \dots, N \\ \frac{\partial L}{\partial \bar{u}_{i,k}} & \quad i = 1, 2, \dots, r; \quad k = 0, 1, 2, \dots, N-1 \\ \frac{\partial L}{\partial \bar{\lambda}_{i,k}} & \quad i = 1, 2, \dots, n; \quad k = 1, 2, \dots, N \end{aligned}$$

Usando una forma vectorial como abreviación:

$$\begin{aligned} \frac{\partial L}{\partial \bar{\mathbf{x}}_k} & \quad k = 1, 2, \dots, N \\ \frac{\partial L}{\partial \bar{\mathbf{u}}_k} & \quad k = 0, 1, 2, \dots, N-1 \\ \frac{\partial L}{\partial \bar{\lambda}_k} & \quad k = 1, 2, \dots, N \end{aligned}$$

Recordando los resultados para diferenciar formas complejas cuadráticas y bilineales con respecto a variables vectoriales

$$\frac{\partial \mathbf{x}^* A \mathbf{x}}{\partial \bar{\mathbf{x}}} = A \mathbf{x} \quad \frac{\partial \mathbf{x}^* A \mathbf{y}}{\partial \bar{\mathbf{x}}} = A \mathbf{y}$$

$$\frac{\partial L}{\partial \bar{\mathbf{x}}_k} = Qx_k + G^* \lambda_{k+1} - \lambda_k = 0 \quad k = 1, 2, \dots, N-1 \quad (\text{B.6})$$

$$\frac{\partial L}{\partial \bar{\mathbf{x}}_N} = Sx_N - \lambda_N = 0 \quad (\text{B.7})$$

$$\frac{\partial L}{\partial \bar{\mathbf{u}}_k} = Ru_k + H^* \lambda_{k+1} = 0 \quad k = 0, 1, 2, \dots, N-1 \quad (\text{B.8})$$

$$\frac{\partial L}{\partial \bar{\lambda}_k} = Gx_{k-1} + Hu_{k-1} - x_k = 0 \quad k = 1, 2, \dots, N \quad (\text{B.9})$$

El vector λ_k es llamado vector adjunto o covector. Para resolver la minimización hay que simplificar las ecuaciones. De la ecuación (B.6) calculamos

$$\lambda_k = Qx_k + G^* \lambda_{k+1}, \quad k = 1, 2, \dots, N-1 \quad (\text{B.10})$$

$\lambda_N = Sx_N$ se obtiene de la ecuación (B.7). Considerando que R^{-1} existe, obtenemos u_k de la ecuación (B.8).

$$u_k = -R^{-1}H^* \lambda_{k+1}, \quad k = 0, 1, 2, \dots, N-1 \quad (\text{B.11})$$

La ecuación (B.9) corresponde a la ecuación de estado del sistema y puede reescribirse como:

$$x_{k+1} = Gx_k + Hu_k \quad (\text{B.12})$$

Llevando (B.11) a (B.12)

$$x_{k+1} = Gx_k - HR^{-1}H^* \lambda_{k+1} \quad (\text{B.13})$$

con la condición inicial $x_0 = 0$. Para resolver este problema de minimización hay que resolver simultáneamente las ecuaciones (B.10) y (B.13). La primera tiene condición final λ_N y la segunda tiene condición inicial. La solución de este sistema utilizando la transformación de Riccati corresponde a un sistema de lazo cerrado o realimentado. La transformación de Riccati consiste en suponer que λ_k puede escribirse como:

$$\lambda_k = P_k x_k \quad (\text{B.14})$$

donde P_k es una matriz hermitiana de dimensión $n \times n$

Llevando (B.14) a (B.10) obtenemos:

$$P_k x_k = Q x_k + G^* P_{k+1} x_{k+1} \quad (\text{B.15})$$

Llevando (B.14) a (B.12) y reagrupando términos obtenemos,

$$x_{k+1} = [I + HR^{-1}H^*P_{k+1}]^{-1}Gx_k \quad (\text{B.16})$$

Sustituyendo la expresión anterior en la ecuación (B.15) y reagrupando términos,

$$[P_k - Q - G^* P_{k+1} [I + HR^{-1}H^*P_{k+1}]^{-1}G]x_k = 0 \quad (\text{B.17})$$

Debido a que la ecuación (B.17) debe cumplirse para todo x_k

$$P_k = Q + G^* P_{k+1} [I + HR^{-1}H^*P_{k+1}]^{-1}G \quad (\text{B.18})$$

Usando el lema de inversión de matrices

$$(A + BD)^{-1} = A^{-1} - A^{-1}B(I + DA^{-1}B)^{-1}DA^{-1} \quad (\text{B.19})$$

y haciendo

$$\begin{aligned} A &= I \\ B &= HR^{-1} \\ D &= H^*P_{k+1} \end{aligned}$$

calculamos

$$(I + HR^{-1}H^*P_{k+1})^{-1} = I - H(R + H^*P_{k+1}H)^{-1}H^*P_{k+1} \quad (\text{B.20})$$

llevando el resultado anterior a la ecuación (B.18)

$$\begin{aligned} P_k &= Q + G^* P_{k+1} [I - H(R + H^*P_{k+1}H)^{-1}H^*P_{k+1}]G \\ P_k &= Q + G^* P_{k+1}G - G^* P_{k+1}H(R + H^*P_{k+1}H)^{-1}H^*P_{k+1}G \end{aligned} \quad (\text{B.21})$$

La ecuación (B.21), o su equivalente la ecuación (B.18), son llamadas ecuaciones algebraicas de Riccati.

Para la función de retroalimentación usamos la ecuación (B.11). Para otras formas equivalentes de la función de retroalimentación consultar [Oga95]. Primero

encontramos una expresión equivalente para λ_{k+1} . Usando la ecuaciones (B.14) y (B.10)

$$\begin{aligned}\lambda_k &= Qx_k + G^* \lambda_{k+1} \\ P_k x_k &= Qx_k + G^* \lambda_{k+1} \\ \lambda_{k+1} &= (G^*)^{-1}[P_k - Q]x_k\end{aligned}\tag{B.22}$$

Usando la ecuación (B.18) en la ecuación (B.22)

$$\begin{aligned}\lambda_{k+1} &= (G^*)^{-1}(Q + G^* P_{k+1}[I + HR^{-1}H^* P_{k+1}]^{-1}G - Q)x_k \\ &= (G^*)^{-1}(G^* P_{k+1}[I + HR^{-1}H^* P_{k+1}]^{-1}G)x_k \\ &= (((P_{k+1})^{-1})^{-1}[I + HR^{-1}H^* P_{k+1}]^{-1})Gx_k \\ &= [(P_{k+1})^{-1} + HR^{-1}H^*]^{-1}Gx_k\end{aligned}\tag{B.23}$$

Llevando el resultado (B.23) a la ecuacion (B.11)

$$u_k = -R^{-1}H^*[(P_{k+1})^{-1} + HR^{-1}H^*]^{-1}Gx_k\tag{B.24}$$

Haciendo $u_k = -K_k x_k$, y considerando el resultado (B.24), obtenemos K_k

$$K_k = R^{-1}H^*[(P_{k+1})^{-1} + HR^{-1}H^*]^{-1}G\tag{B.25}$$

Una expresión equivalente para K_k es $[R + H^* P_{k+1} H]^{-1} H^* P_{k+1} G$. Para verlo calculamos $[R + H^* P_{k+1} H]^{-1} H^* P_{k+1} [(P_{k+1})^{-1} + HR^{-1}H^*]$

$$\begin{aligned}& [R + H^* P_{k+1} H]^{-1} H^* P_{k+1} [(P_{k+1})^{-1} + HR^{-1}H^*] \\ & [R + H^* P_{k+1} H]^{-1} H^* [I + P_{k+1} + HR^{-1}H^*] \\ & [R + H^* P_{k+1} H]^{-1} [H^* + H^* P_{k+1} + HR^{-1}H^*] \\ & [R + H^* P_{k+1} H]^{-1} [(RR^{-1})H^* + H^* P_{k+1} + HR^{-1}H^*] \\ & [R + H^* P_{k+1} H]^{-1} [RR^{-1}H^* + H^* P_{k+1} + HR^{-1}H^*] \\ & [R + H^* P_{k+1} H]^{-1} [R + H^* P_{k+1} + H]R^{-1}H^* = R^{-1}H^*\end{aligned}$$

Por lo que

$$[R + H^* P_{k+1} H]^{-1} H^* P_{k+1} [(P_{k+1})^{-1} + HR^{-1}H^*] = R^{-1}H^*\tag{B.26}$$

Multiplicando ambos lados de la ecuación (B.26) por $[(P_{k+1})^{-1} + HR^{-1}H^*]^{-1}$

otenenmos

$$[R + H^* P_{k+1} H]^{-1} H^* P_{k+1} = R^{-1} H^* [(P_{k+1})^{-1} + H R^{-1} H^*]^{-1} \quad (\text{B.27})$$

Llevando este resultado a la ecuación (B.25) obtenemos una expresión equivalente para K_k

$$K_k = [R + H^* P_{k+1} H]^{-1} H^* P_{k+1} G \quad (\text{B.28})$$

Para realizar un diagrama de bloques usamos la ecuación (B.12) y obtenemos el sistema realimentado de la Figura B.1. K_k se calcula según la ecuación (B.28).

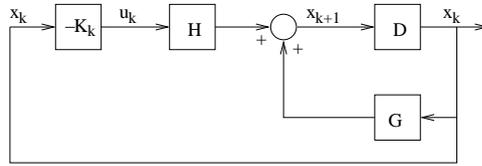


Figura B.1: Sistema realimentado LQR de horizonte finito (Linear quadratic regulator).

Cuando el proceso de control cuadrático óptimo es finito (cuando N es finito), la matriz K_k varía según la ecuación (B.28). Cuando $N \rightarrow \infty$, la solución del óptimo del problema de control cuadrático pasa al estado estable. En este estado, K_k pasa a ser una matriz constante K . También P_k se convierte en una matriz constante. Usando la ecuación (B.21) tenemos

$$P = Q + G^* P G - G^* P H (R + H^* P H)^{-1} H^* P G \quad (\text{B.29})$$

Llevando este resultado a la ecuación (B.28), obtenemos K en estado estable

$$K = [R + H^* P H]^{-1} H^* P G \quad (\text{B.30})$$

Como resumen del control lineal cuadrático en estado estable, listamos las ecua-

ciones para un sistema lineal con incógnita x aún cuando puedan ser repetidas:

$$x_{k+1} = Gx_k + Hu_k \quad (\text{B.31})$$

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [x_k^* Q x_k + u_k^* R u_k] \quad (\text{B.32})$$

$$u_k = -K r_k \quad (\text{B.33})$$

$$r_k = b - Ax_k \quad (\text{B.34})$$

$$K = [R + H^* P H]^{-1} H^* P G \quad (\text{B.35})$$

$$P = Q + G^* P G - G^* P H (R + H^* P H)^{-1} H^* P G \quad (\text{B.36})$$

B.2 Condiciones de estabilidad de un controlador LQR

Las condiciones de estabilidad de un controlador lineal cuadrático están indicadas en el Teorema 1 que está demostrado en [Ber05]. Hemos cambiado los nombres de las matrices del teorema original para que puedan ser fácilmente relacionadas con la deducción de la sección anterior.

Teorema 1 (Condiciones de estabilidad de un controlador LQR [Ber05]). *Sea G una matriz de dimensión $n \times n$, H una matriz de dimensión $n \times m$, Q una matriz positiva semidefinida de dimensión $n \times n$, R una matriz simétrica positiva definida de dimensión $m \times m$. Considerar la ecuación algebraica discreta de Riccati*

$$P_{k+1} = Q + G^* P_k G - G^* P_k H (R + H^* P_k H)^{-1} H^* P_k G, \quad k = 0, 1, 2, 3, \dots \quad (\text{B.37})$$

en la que P_0 es una matriz arbitraria simétrica positiva semidefinida. Suponer que el par (G, H) es controlable. Suponer que Q puede ser escrito $Q = CC^*$, donde el par (G, C) es observable. Entonces:

- Existe una matriz P simétrica positiva definida, tal que para toda matriz inicial P_0 simétrica positiva semidefinida se cumple

$$\lim_{k \rightarrow \infty} P_k = P \quad (\text{B.38})$$

Además, P es la única solución de la ecuación algebraica de Riccati

$$P = Q + G^* P G - G^* P H (R + H^* P H)^{-1} H^* P G, \quad k = 0, 1, 2, 3, \dots$$

en la clase de matrices simétricas positivas semidefinidas.

- *El sistema de lazo cerrado ($x_{k+1} = Gx_k - HKx_k$) correspondiente es estable: los autovalores de la matriz $G - HK$ están estrictamente dentro del círculo unitario. K es la matriz*

$$K = [R + H^*PH]^{-1}H^*PG$$

Demostración. Ver la proposición 4.4.1 en [Ber05] □

Las suposiciones de la proposición anterior pueden ser relajadas. El par (G, H) se considera estabilizable en lugar de controlable. Estabilizable significa que existe B tal que el sistema $x_{k+1} = (G + HB)x_k$ es estable. Por otra parte, el par (G, C) es detectable en lugar de observable. Detectable significa que si $u_k \rightarrow 0$ y $Cx_k \rightarrow 0$ entonces $x_k \rightarrow 0$, que equivale a decir que la inestabilidad del sistema puede detectarse comparando los valores de la secuencia $z_k = Cx_k$. Entonces reemplazando las suposiciones de que el par (G, H) es controlable por estabilizable y que (G, C) es observable por detectable, la proposición anterior se cumple solo que P pasa a ser positiva semidefinida en lugar de positiva definida.[Ber05]

B.3 LQRES y LQRES D

Helmke et al. utilizan teoría de Sistemas lineales con costo cuadrático (también conocido como LQR Linear Quadratic Control) para formular la resolución de $Ax = b$ y diseñar un algoritmo que minimiza el residuo [HJL06].

Helmke et al. utilizan una iteración de Richardson a la cual agregan el término de control $BK(b - Ax_k)$

$$x_{k+1} = (I - A)x_k + BK(b - Ax_k) + b \tag{B.39}$$

Calculan el residuo $r_{k+1} = b - Ax_{k+1}$ a partir de la ecuación (B.39) y obtienen el sistema (B.40) *donde la incógnita es el residuo*. Helmke et al. la teoría de Control Lineal Cuadrático aplican al sistema (B.40). Utilizan la norma euclidiana para el funcional de la ecuación (B.3) por lo que $Q = I_n$ y $R = I_m$ son las matrices identidad de dimensión n y m respectivamente. Calculan el residuo como un sistema con costo cuadrático en estado estable. Es decir, buscan que el residuo esté tan próximo a cero como sea posible.

$$r_{k+1} = (I - A)r_k - ABu_k \tag{B.40}$$

Utilizando $I - A = G$, $-AB = H$, $I_n = Q$ y $I_m = R$ en las ecuaciones para

resolver el sistema del residuo en estado estable obtenemos el sistema dinámico, que también puede ser visto en el diagrama de bloques de la Figura B.2

$$\begin{aligned}
 r_{k+1} &= (I - A)x_k - ABu_k \\
 u_k &= -Kr_k \\
 r_k &= b - Ax_k \\
 K &= [I_m - (AB)^*PAB]^{-1}(AB)^*P(I - A) \\
 P &= I_n + (I - A)^*P(I - A) \\
 &\quad + (I - A)^*(AB)(I_m + (AB)^*P(AB))^{-1}(AB)^*P(I - A)
 \end{aligned}$$

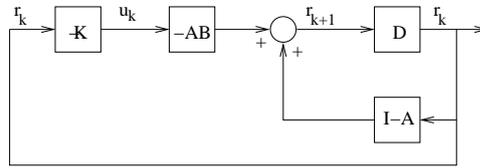


Figura B.2: Residuo como Sistema realimentado LQR en estado estable.

Notar que K y P no dependen de la iteración y necesitan ser calculados una sola vez. Con estos resultados construyen el método iterativo LQRES [HJL06].

Entrada: A matriz del sistema, B parámetro del método, r_0 residuo inicial (comúnmente $x_0 = 0$, por lo que $r_0 = b$), $maxIt$ número de iteraciones, *tolerancia* Si se cumple $\|r\|/\|r_0\| < tolerancia$ se considera que el algoritmo convergió

Salida: x resultado calculado

$$K = [I_m - (AB)^*PAB]^{-1}(AB)^*P(I - A)$$

$$\text{Resolver } P = I_n + (I - A)^*P(I - A)$$

$$+ (I - A)^*(AB)(I_m + (AB)^*P(AB))^{-1}(AB)^*P(I - A)$$

$$iteracion = 0$$

mientras ($\|r\|/\|r_0\| < tolerancia$ and $iteracion < maxIt$) **hacer**

$$u_k = -Kr_{iteracion}$$

$$x_{iteracion+1} = (I - A)x_{iteracion} + BKr_{iteracion} + b$$

$$r_{iteracion+1} = (I - A)x_k - ABu_k$$

$$iteracion = iteracion + 1$$

fin_mientras

si $\|r_{iteracion}\|/\|r_0\| < tolerancia$ **entonces**

retornar: $x_{[iteracion]}$

fin_si

Algoritmo 7: LQRES

Considerando que la resolución exacta de la ecuación algebraica de Riccati para obtener P es costosa, Helmke et. al proponen utilizar una solución aproximada de P [HJ10]. Este algoritmo es llamado $LQRES$. Para calcular la solución aproximada de P puede utilizarse el Teorema 1. Es decir, calcular P_{k+1} usando la relación $P_{k+1} = Q + G^*P_kG - G^*P_kH(R + H^*P_kH)^{-1}H^*P_kG$. Este algoritmo es llamado $LQRES$, la única diferencia con respecto a $LQRES$ es el cálculo aproximado de P . P_0 puede ser cualquier matriz definida positiva, en particular usamos $P_0 = I$. Para $M = 2$, se calcula P_1 usando P_0 y luego P_2 usando P_1 .

A continuación incluimos algunas pruebas con $LQRES$ y $LQRES$ (M).

B.4 Experimentos

B.4.1 Experimentos con matriz de pequeñas dimensiones

La matriz utilizada fue presentada originalmente por Embreé [Emb03]. Estos experimentos reproducen los resultados obtenidos por Helmke [HJ10]. El sistema lineal $Ax = b$ probado tiene las matrices:

$$A = \begin{bmatrix} 1 & 2 & -2 \\ 0 & 2 & 4 \\ 0 & 0 & 3 \end{bmatrix} \quad b = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$$

El método LQRES introduce como parámetro una matriz B que debe ser elegida de tal forma que el para $(I - A, -AB)$ sea estabilizable. Los valores usados para B son [HJ10]

$$B1 = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \quad B2 = \begin{bmatrix} 3 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \quad B3 = \begin{bmatrix} 3 & -1 \\ 1 & -2 \\ 1 & -3 \end{bmatrix}$$

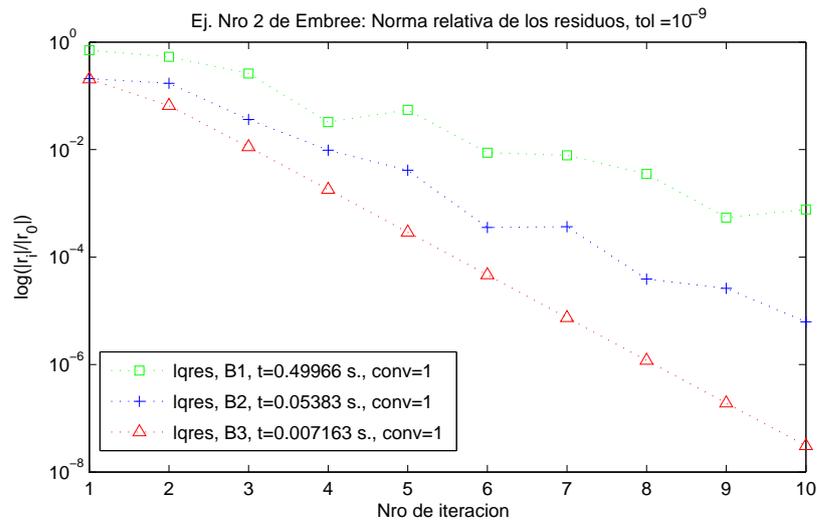


Figura B.3: LQRES: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

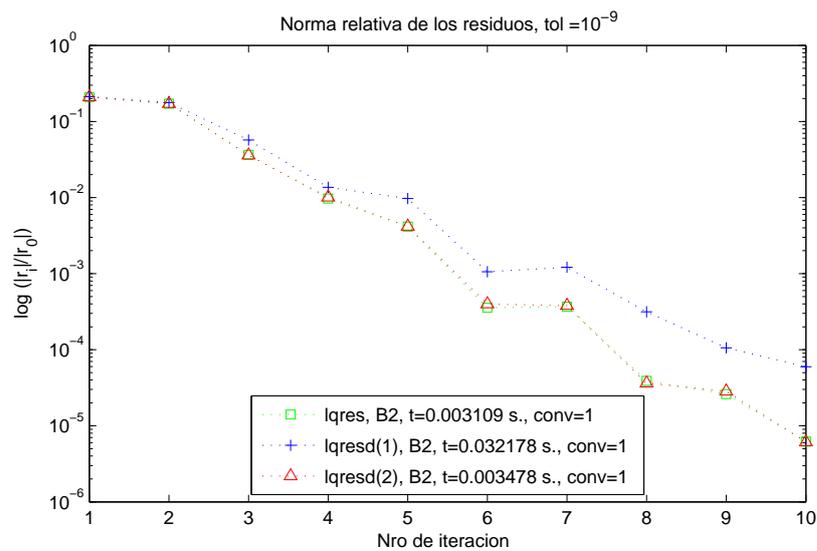


Figura B.4: LQRES y LQRES(D(M)): $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

B.4.2 Experimentos con matrices de problemas reales

En la práctica, encontrar el parámetro B de LQRES(D(M)) no es fácil. Helmke et al. [?] indican que se puede usar cualquier matriz invertible. Para los ejemplos usamos la matriz identidad I del tamaño adecuado al del sistema. Usar directamente el lado derecho b en general no funcionó. Para la matriz Young3c tanto LQRES como LQRES(D(M)) convergen. Sin embargo, para las matrices Mahindas y Shl_400 ambos no convergen.

Matriz Young3c

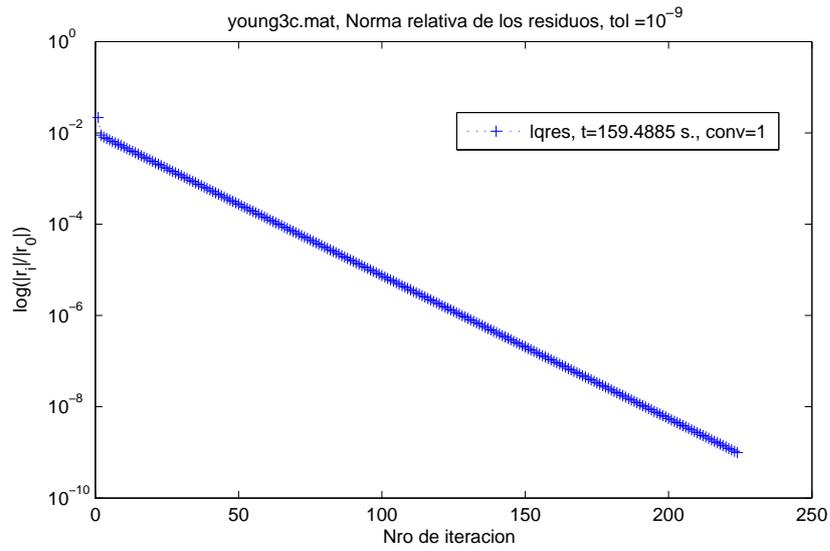


Figura B.5: LQRES: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

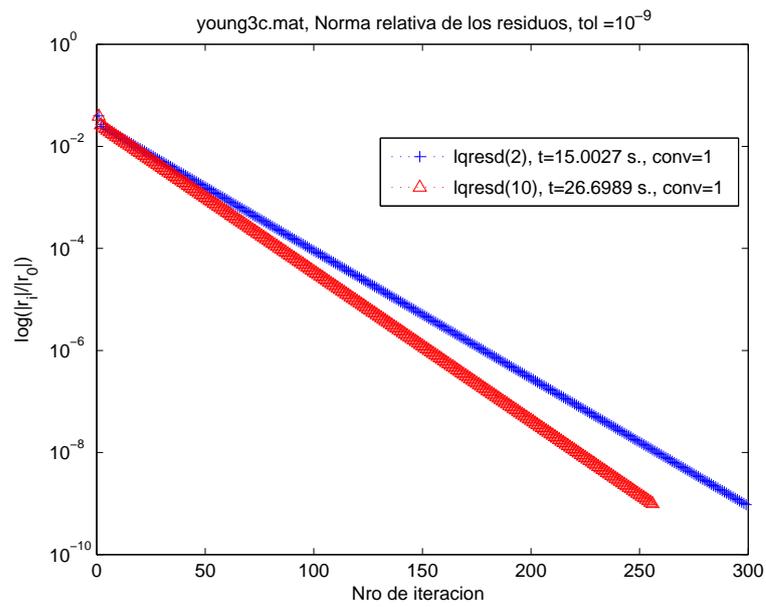


Figura B.6: LQRES(D(M)): $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

Matriz Mahindas

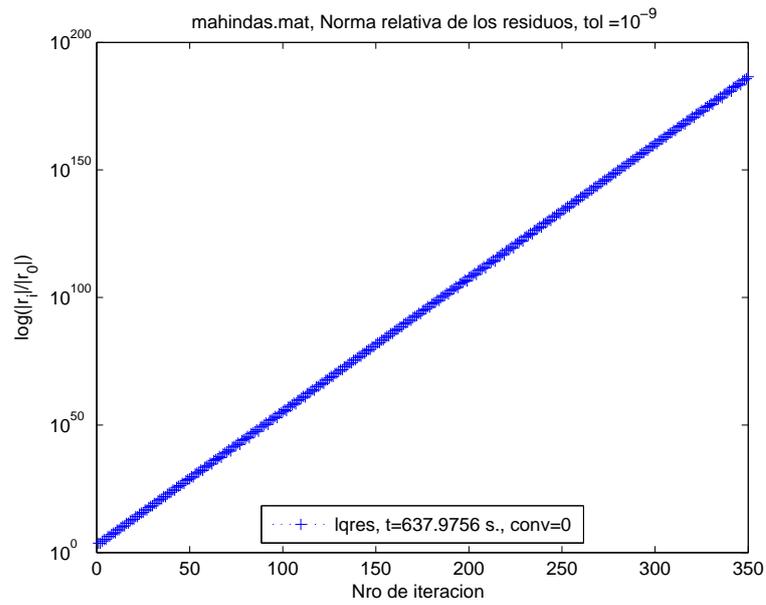


Figura B.7: LQRES: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

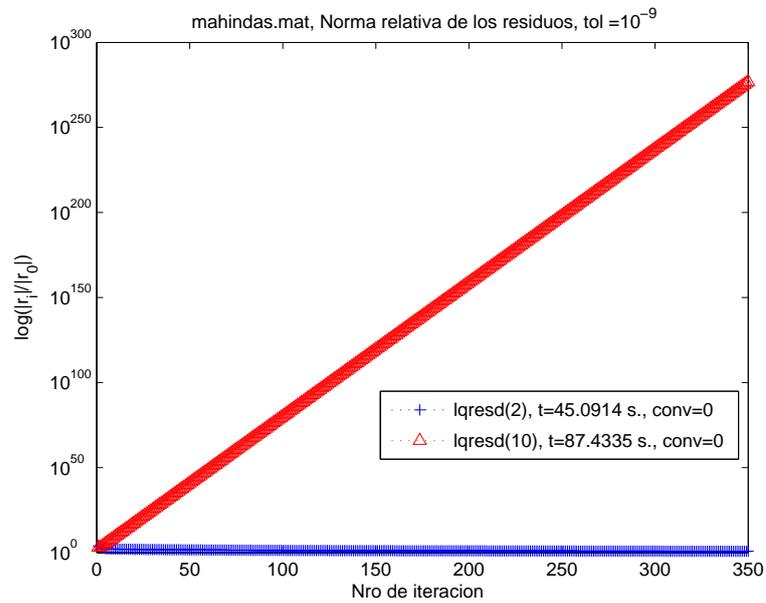


Figura B.8: LQRES(D(M)): $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

Matriz Shl_400

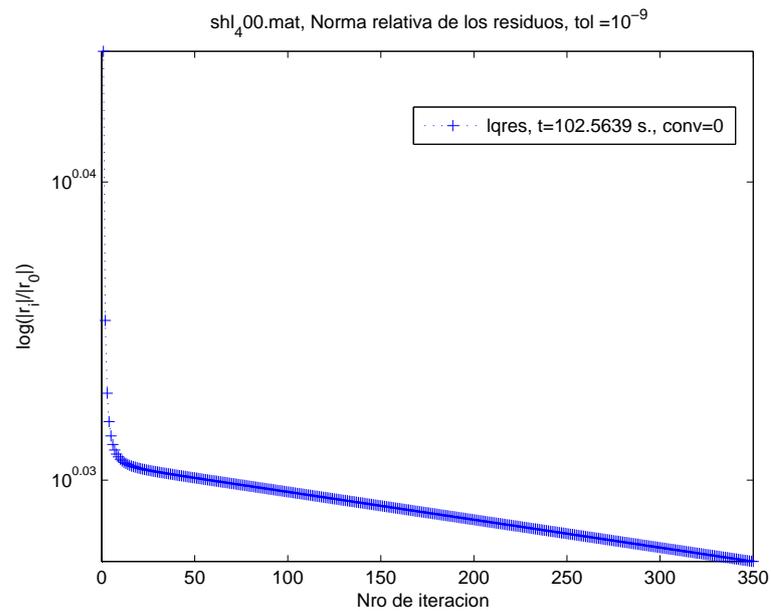


Figura B.9: LQRES: $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

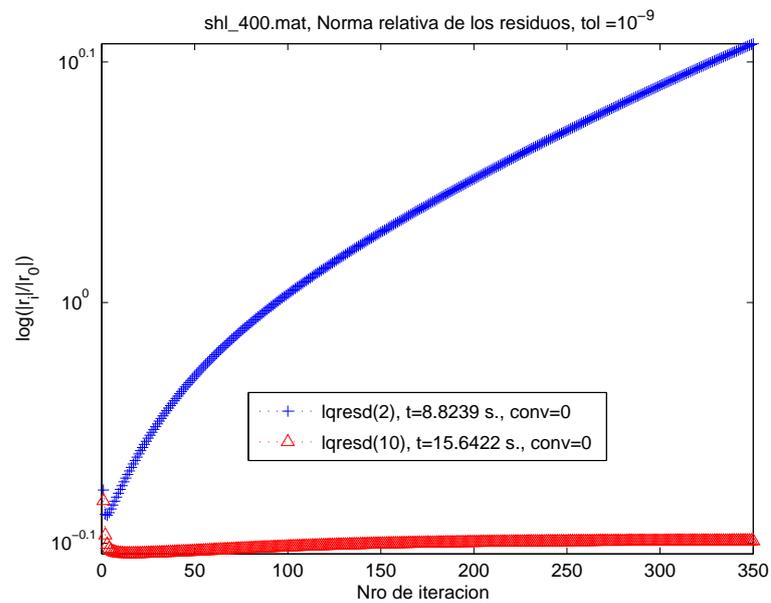


Figura B.10: LQRES(D(M)): $\log(\|r_j\|/\|r_0\|)$ Vs Iteraciones

B.5 Conclusión

LQRES es una alternativa para GMRES, que minimiza el residuo y si se cumplen las condiciones del controlador LQR es un método estable. Sin embargo, tiene dos problemas:

- el cálculo de P que es la solución de una ecuación algebraica de Riccati, este cálculo es costoso en general más caro que la eliminación Gaussiana [BK04].
- la elección del parámetro B que es una matriz o vector. Encontrar algún B no es trivial para matrices no pequeñas.

Con LQRES(D,M) Helmke et al. de cierta forma resuelven el primer punto, pero la elección de B continua siendo problemática. En los ejemplos se muestran dos casos en los LQRES(D,M) no converge, puede argumentarse que $B = I$ no satisface la condición de estabilidad, pero para problemas de simulación B no puede ser generado de forma aleatoria porque consumiría bastante tiempo. (Helmke et al. generan B de tamaño 50×3 de forma aleatoria para una matriz A de tamaño 50×50 también generada aleatoriamente [HJ10]).