Universidad de Valladolid

Facultad de Ciencias

Trabajo fin de grado en matemáticas

EL MÉTODO DE ACELERACIÓN DE ANDERSON

Autor: Víctor R. Díez Recio
Tutor: Ángel Durán Martín

10 de diciembre de 2024

Resumen

La aceleración de Anderson (AA) es un método utilizado para acelerar la convergencia en problemas iterativos, donde las técnicas clásicas son insuficientes. Este trabajo aborda el análisis detallado de AA, su conexión con algoritmos como GMRES y FOM en el caso lineal y su relación con los llamados métodos multisecantes en escenarios no lineales. Además, se presentan resultados numéricos que evidencian su rendimiento en problemas lineales y no lineales.

Palabras clave: Aceleración de Anderson, iteraciones de punto fijo, métodos de Krylov, GMRES.

Abstract

Anderson Acceleration (AA) is a method used to accelerate convergence in iterative problems, where classical techniques are insufficient. This work presents a detailed analysis of AA, its connection with algorithms such as GMRES and FOM in the linear case, and its relationship with so-called multisecant methods in nonlinear scenarios. Additionally, numerical results are provided, showing its performance in both linear and nonlinear problems.

Key words: Anderson Acceleration, fixed-point iterations, Krylov methods, GMRES.

Índice general

Lista de gráficos

Algoritmos

Capítulo 1

Introducción

La cuestión principal sobre la que trata este trabajo se centra en las iteraciones de punto fijo y qué métodos son los mejores para acelerar su convergencia. Consideramos un problema general de punto fijo

$$\text{“dado } g\colon \mathbb{R}^n \to \mathbb{R}^n, \text{ resolver } x = g(x)\text{”}. \tag{1.0.1}$$

El problema (1.0.1) puede reformularse como un problema de resolución de ecuaciones algebraicas

$$\text{“dado } f\colon \mathbb{R}^n \to \mathbb{R}^n, \text{ resolver } f(x) = 0\text{”}. \tag{1.0.2}$$

La relación entre las ecuaciones (1.0.1) y (1.0.2) se establece mediante la deducción de una función f a partir de g, o viceversa, bajo la condición de que x es un punto fijo de g si, y solo si, es una raíz de f. No siendo, en general, posible la resolución exacta ni de (1.0.1) ni de (1.0.2), es necesario buscar una aproximación mediante algún procedimiento numérico. Un método clásico se basa en la búsqueda de aproximaciones de forma iterativa, generando una sucesión que, bajo ciertas condiciones, puede converger a la solución del problema. Las técnicas elementales que se explican en el Grado de Matemáticas basadas en esta idea, [SS], pueden formularse a partir del problema (1.0.1) o de (1.0.2). Entre estas se encuentran el algoritmo de bisección (para el caso escalar $n = 1$), el método de la secante (y sus versiones para $n > 1$, como el método de Broyden, [B]), la iteración de punto fijo clásica

$$x_{k+1} = g(x_k), \quad k = 0, 1, \ldots,$$

o el método de Newton

$$\begin{aligned} x_{k+1} &= x_k + \delta_k, \\ J_f(x_k)\delta_k &= -f(x_k), \quad k = 0, 1, \ldots, \end{aligned}$$

donde $J_f(x)$ denota el jacobiano de f en x,

$$J_f(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix},$$

y que puede formularse como iteración de punto fijo, con función de iteración

$$g(x) = x - J_f^{-1}(x) f(x),$$

donde asumimos f diferenciable.

1.1. Métodos de aceleración

Todos estos procedimientos requieren, para su convergencia a una solución de (1.0.1) o (1.0.2), de condiciones más o menos restrictivas. Centrados en iteraciones de punto fijo, el mayor problema se encuentra en que las iteraciones pueden no converger y, de hacerlo, solo con velocidad lenta. Es aquí es donde entran los métodos de aceleración, que pueden potencialmente aliviar la convergencia lenta y, en algunos casos, también la divergencia. El objetivo de estos métodos es transformar una sucesión de vectores generada por algún proceso en una nueva sucesión que converja más rápido que la sucesión inicial. Los más populares se pueden clasificar en dos categorías: los métodos polinómicos y los algoritmos tipo ε. La primera familia incluye métodos como el del polinomio mínimo (MPE) de Cabay y Jackson, el método de rango reducido (RRE) de Eddy y Mesina, y el método de polinomio mínimo modificado (MMPE) de Sidi et al, Brezinski y Pugachev. La segunda clase incluye el algoritmo topológico ε (TEA) de Brezinski y los algoritmos ε escalares y vectoriales (SEA y VEA) de Wynn. Los métodos polinómicos tienen origen en la aceleración de la convergencia para la resolución de sistemas lineales, [JS], y su formulación se extendió más adelante al caso de sistemas no lineales, [SI]. Por su parte, los algoritmos de tipo ε fueron inicialmente formulados para el caso escalar no lineal y extendidos después a sistemas, [BR].

1.2. Presentación del método de aceleración de Anderson

Nuestro interés en este trabajo radica en un método de aceleración particular al que nos referimos como método de aceleración de Anderson, el cual denotaremos a menudo por AA. El método se implementa en el algoritmo 1, donde $\|\cdot\|_2$ representa la norma euclídea.

Esta técnica fue propuesta por Donald G. Anderson en 1965, [A], como procedimiento para evitar algunas desventajas de los métodos clásicos de iteración de punto fijo en sistemas algebraicos generados en la resolución numérica de ecuaciones integrales no lineales, pero en su trabajo original, ya sugiere que el procedimiento puede generalizarse a otros sistemas de ecuaciones no lineales.

La aceleración de Anderson difiere matemáticamente de los métodos polinómicos y ε-algoritmos y, de hecho, pertenece a una categoría distinta de métodos.

AA destaca por su versatilidad y por tener un buen rendimiento en situaciones difíciles. Por eso es muy utilizada en simulaciones en dinámica molecular, física computacional, biofísica, bioinformática, problemas de dinámica de fluidos computacional, entre muchos otros. Recientemente, el profesor de Oxford Samar Khatiwala, ha publicado un artículo sobre la simulación de modelos del sistema terrestre, los cuales son fundamentales para predecir el cambio climático futuro, donde la aceleración de Anderson tiene un papel vital para ahorrar tiempo computacional, [K].

Nuestro objetivo en este trabajo es profundizar y analizar con detalle la aceleración de Anderson y su comportamiento, así como su equivalencia con otros algoritmos. En el capítulo 2, nos centraremos en introducir los algoritmos de Arnoldi (algoritmo 2) y GMRES (algoritmo 4), y estudiar la equivalencia entre este último y la aceleración de Anderson en problemas lineales, así como su convergencia. En el capítulo 3, revisamos la relación de la aceleración de Anderson con los llamados métodos multisecante, [FS]. Además, mostramos que un miembro particular de la familia de métodos de Anderson (el método Tipo I) es, en problemas lineales, esencialmente equivalente al método de Arnoldi o FOM (Algoritmo 2). En ambos capítulos,

ilustraremos los resultados con algunos ejemplos numéricos.

Algoritmo 1: Aceleración de Anderson

Entrada

- $x_0 \in \mathbb{R}^n$*: Solución aproximada inicial.*
- $g : \mathbb{R}^n \to \mathbb{R}^n$*: Función de iteración.*
- $m \in \mathbb{Z}$*: Truncamiento, que será mayor o igual que* 1*.*

Salida

- $x_{k+1} \in \mathbb{R}^n$*: Solución aproximada del sistema.*

Función `AA`(x_0, g, m)**:**

$x_1 \leftarrow g(x_0)$;
para $k \leftarrow 1, 2, \ldots$ **hacer**
$m_k \leftarrow \min(k, m)$;
$F_k \leftarrow \begin{bmatrix} f_{k-m_k} & \cdots & f_k \end{bmatrix}$, donde $f_i \leftarrow g(x_i) - x_i$;
Determinar $\alpha^{(k)} = \left(\alpha_0^{(k)}, \ldots, \alpha_{m_k}^{(k)}\right)^T$ que resuelve

$$\min_{\alpha=(\alpha_0,\ldots,\alpha_{m_k})^T} \|F_k\alpha\|_2 \quad \text{sujeto a} \quad \sum_{i=0}^{m_k} \alpha_i = 1; \tag{1.2.1}$$

$x_{k+1} \leftarrow \sum_{i=0}^{m_k} \alpha_i^{(k)}\, g(x_{k-m_k+i})$;
devolver x_{k+1};

Capítulo 2

El método AA en problemas lineales

En este capítulo exploraremos el rendimiento del método de aceleración de Anderson (AA) aplicado a problemas lineales, es decir, a la familia de problemas definida en (1.0.1) para el caso específico en que la función de iteración g toma la forma

$$g(x) = Ax + b, \qquad \text{con} \quad A \in \mathbb{R}^{n\times n}, \quad b \in \mathbb{R}^n. \tag{2.0.1}$$

En este caso, el método de aceleración de Anderson tiene gran relación con los métodos de Krylov empleados en la resolución iterativa de sistemas lineales, en particular, con dos de los algoritmos más representativos de esta clase: el Método de Arnoldi o Full Orthogonalization Method (FOM) y el Generalized Minimal Residual Method (GMRES).

En la primera parte del capítulo revisaremos la familia de métodos de Krylov, para luego enfocarnos en FOM y GMRES, explorando tanto su construcción como su convergencia. Posteriormente, analizaremos cómo el método de aceleración de Anderson es esencialmente equivalente, en cierto sentido que especificaremos, a estos últimos en problemas (1.0.1) y (2.0.1).

Para concluir, presentaremos los resultados de convergencia específicos del método AA cuando se aplica a problemas lineales de la forma descrita en (2.0.1), poniendo especial énfasis en la eficiencia del método en este contexto y en la comparación de su rendimiento con los métodos de Krylov mencionados.

2.1. Métodos de Arnoldi y GMRES:

Por su relación con AA, estudiamos aquí los métodos iterativos de Arnoldi y GMRES para la resolución de sistemas lineales

$$Ax = b, \qquad \text{con} \quad A \in \mathbb{R}^{n\times n} \quad \text{no singular y} \quad b \in \mathbb{R}^n. \tag{2.1.1}$$

Los sistemas (2.1.1) siempre han sido objeto de gran estudio en el ámbito matemático, debido a la cantidad de aplicaciones prácticas en las que aparecen. Sin embargo, cuando estos sistemas tienen dimensiones muy grandes o están mal acondicionados, su resolución se vuelve muy compleja y costosa. Una matriz se considera mal acondicionada cuando pequeñas perturbaciones en los datos de entrada resultan en cambios grandes en la solución, indicando inestabilidad en el proceso de resolución.

Aunque los métodos directos teóricamente dan la solución exacta de (2.1.1) en un número finito de operaciones, su aplicación en sistemas de gran tamaño y con estructura está limitada por diversos problemas computacionales, como la propagación de errores de redondeo, la capacidad de almacenamiento y el mal acondicionamiento. Por su parte, los métodos iterativos,

que buscan la solución de (2.1.1) a partir de un proceso de convergencia con aproximaciones sucesivas, suelen ser una alternativa más eficiente en este contexto, al aprovechar, sobre todo, la estructura de la matriz. Si bien estos métodos no garantizan una solución exacta en un número finito de pasos, son más adecuados para problemas de gran escala.

Hay varios procedimientos para el diseño de métodos iterativos. Los más básicos, como el de Jacobi, Gauss-Seidel o el de sobrerelajación, se basan en la modificación conveniente de componentes de la solución aproximada y generan una sucesión que, bajo ciertas condiciones, converge a la solución de (2.1.1), [S].

Uno de los enfoques más actuales consiste en el uso de los llamados métodos de proyección. Estas técnicas buscan una solución aproximada dentro de un subespacio $\mathcal{K} \subseteq \mathbb{R}^n$ de dimensión k, imponiendo k condiciones independientes de ortogonalidad sobre el residuo $b - Ax$, que definen un subespacio de restricciones $\mathcal{L}$ de dimensión k. Entre los métodos más destacados de proyección están aquellos basados en subespacios de Krylov, donde el subespacio $\mathcal{K}$ se define a partir de un iterante inicial $x_0 \in \mathbb{R}^n$, como

$$\mathcal{K} = \mathcal{K}_k(A, r_0) = span\{r_0, Ar_0, \ldots, A^{k-1}r_0\}, \qquad \text{con} \quad r_0 = b - Ax_0, \quad \text{y} \quad x_0 \in \mathbb{R}^n.$$

La diferencia entre los distintos métodos de Krylov se encuentra principalmente en la selección del subespacio de restricciones $\mathcal{L}$. En este capítulo, nos centraremos en dos de los más conocidos, el método de Arnoldi o Full Orthogonalization Method (FOM) y el llamado Generalized Minimal Residual Method (GMRES), para luego establecer su relación con el método de aceleración de Anderson (AA).

2.1.1. Métodos de proyección

Un método de proyección sobre un subespacio $\mathcal{K}$ y ortogonal a otro subespacio $\mathcal{L}$ es un proceso que encuentra una solución aproximada $\tilde{x}$ de (2.1.1), que satisface

$$\tilde{x} \in \mathcal{K} \quad \text{tal que} \quad b - A\tilde{x} \perp \mathcal{L}.$$

Si deseamos aprovechar el conocimiento de la aproximación inicial x_0 para la solución, el problema debe redefinirse para encontrar una solución aproximada $\tilde{x}$ de (2.1.1), que satisfaga

$$\tilde{x} \in x_0 + \mathcal{K} \quad \text{tal que} \quad b - A\tilde{x} \perp \mathcal{L}. \tag{2.1.2}$$

Si $\tilde{x} = x_0 + \delta$ y $r_0 = b - Ax_0$, entonces $\tilde{x}$ puede calcularse a partir de las condiciones

$$\tilde{x} = x_0 + \delta,$$

$$(r_0 - A\delta, w) = 0, \quad \forall w \in \mathcal{L}. \tag{2.1.3}$$

Los métodos iterativos basados en la proyección consisten en calcular cada aproximación con este procedimiento, volcando en x_0 el último iterante calculado y reiniciando el proceso para la nueva iteración x_k, donde los subespacios $\mathcal{K}$ y $\mathcal{L}$ cambian con k. Más concretamente, a partir de una aproximación inicial x_0, el método genera iteraciones x_k (aproximaciones a la solución de (2.1.1)) en el espacio afín $x_0 + \mathcal{K}_k$ de dimensión k a partir de las condiciones de ortogonalidad sobre el subespacio $\mathcal{L}_k$ de dimensión k

$$x_k \in x_0 + \mathcal{K}_k, \quad b - Ax_k \perp \mathcal{L}_k$$

El método de proyección antes descrito tiene una representación matricial que utilizaremos más adelante. Sea $V = \begin{bmatrix} v_1 & \cdots & v_k \end{bmatrix}$ una matriz $n \times k$ cuyas columnas forman una base de $\mathcal{K} = \mathcal{K}_k$,

y $W = \begin{bmatrix} w_1 & \cdots & w_k \end{bmatrix}$ una matriz $n \times k$ cuyas columnas forman una base de $\mathcal{L} = \mathcal{L}_k$. Utilizando (2.1.2), podemos escribir

$$\tilde{x} = x_0 + Vy,$$

con $y \in \mathbb{R}^k$. Por otro lado, aplicando las condiciones de ortogonalidad en (2.1.3) para $w = w_j$, $j = 1, \ldots, k$, se tiene:

$$W^T AVy = W^T r_0.$$

Si $W^T AV$ es no singular, entonces

$$\tilde{x} = x_0 + V \left(W^T AV\right)^{-1} W^T r_0.$$

En este trabajo, por su relación con el método de aceleración de Anderson, estamos interesados en dos métodos de proyección: el método de Arnoldi o FOM y el método GMRES. El primero es ortogonal ($\mathcal{L} = \mathcal{K}$) y el segundo oblicuo ($\mathcal{L} \neq \mathcal{K}$), pero ambos tienen en común que son métodos basados en subespacios de Krylov.

2.1.2. Métodos de Krylov

Antes de comenzar con los métodos de Krylov, es esencial definir en qué consisten los subespacios en los que se basan, ya que esto ayuda a comprender su naturaleza.

Definición 2.1.1. *Sea $A \in \mathbb{R}^{n\times n}$, $r \in \mathbb{R}^n$, y un entero positivo k, definimos la sucesión de Krylov de orden k como el conjunto de vectores*

$$r, Ar, A^2r, \ldots, A^{k-1}r. \tag{2.1.4}$$

El subespacio de Krylov $\mathcal{K}_k(A, r)$ generado por A y r es

$$span\left\{r, Ar, A^2r, \ldots, A^{k-1}r\right\}.$$

Si los vectores (2.1.4) son linealmente independientes, entonces la dimensión del subespacio $\mathcal{K}_k(A, r)$ será k. En general no podemos garantizar que la dimensión de $\mathcal{K}_k(A, r)$ sea igual a k. De hecho, más adelante veremos que GMRES terminará cuando se produzca $\dim \mathcal{K}_k(A, r) < k$. Definimos a continuación un concepto clave para esta sección.

Definición 2.1.2. *Sea $A \in \mathbb{R}^{n\times n}$ una matriz regular y $r \in \mathbb{R}^n$ un vector. Se llama polinomio mínimo de r respecto de A al polinomio mónico $p \in \mathbb{P}_\gamma$ de menor grado $\gamma \geq 0$ tal que:*

$$p(A)\, r = 0$$

Denominamos a γ como grado de r respecto de A. Lo denotaremos como grado(A, r).

Observación 2.1.1. *Los vectores de $\mathcal{K}_k(A, r)$ pueden ser vistos como los vectores $x \in \mathbb{R}^n$ que pueden escribirse como $x = p(A)r$ con $p(A)$ polinomio de grado menor o igual que $k - 1$. Los coeficientes de $p(A)r$ son los coeficientes de x en la base $\{r, Ar, A^2r, \ldots, A^{k-1}r\}$ de $\mathcal{K}_k(A, r)$.*

Proposición 2.1.2. *Sea $A \in \mathbb{R}^{n\times n}$ una matriz regular, $r \in \mathbb{R}^n$ un vector y $k \geq 1$ natural. Las siguientes afirmaciones son equivalentes:*

(a) $r, Ar, \ldots, A^k r$ son vectores linealmente dependientes.

(b) $\mathcal{K}_k(A, r) = \mathcal{K}_{k+1}(A, r)$.

(c) $A\mathcal{K}_k(A,r) \subseteq \mathcal{K}_k(A,r)$.

(d) $A^{-1}r \in \mathcal{K}_k(A,r)$.

Demostración. (a) $\Rightarrow$ *(b)*: Por la definición 2.1.1, es claro que $\mathcal{K}_k(A,r) \subseteq \mathcal{K}_{k+1}(A,r)$. Por otro lado, de la hipótesis *(a)*, existen $\alpha_0, \alpha_1, \ldots, \alpha_{k-1} \in \mathbb{R}$ no todos nulos tales que $A^k r = \sum_{j=0}^{k-1} \alpha_j A^j r$. Por tanto, se tiene que $A^k r \in \mathcal{K}_k(A,r)$ y $\mathcal{K}_{k+1}(A,r) \subseteq \mathcal{K}_k(A,r)$, por lo que tales subespacios son iguales.

(b) $\Rightarrow$ *(c)*: Por definición de subespacios de Krylov y por *(b)*, obtenemos

$$A\mathcal{K}_k(A,r) \subseteq \mathcal{K}_{k+1}(A,r) = \mathcal{K}_k(A,r).$$

(c) $\Rightarrow$ *(d)*: Sea la aplicación lineal $L\colon \mathcal{K}_k(A,r) \to \mathcal{K}_k(A,r)$ definida por $L(v) = Av$, con $v \in \mathcal{K}_k(A,r)$. Por *(c)*, L está bien definida y, además, es inyectiva por ser A una matriz regular. Por tanto, $\dim(\mathrm{Im}(L)) = \dim(\mathcal{K}_k(A,r))$ y L es una aplicación biyectiva. Luego, existe un vector $v \in \mathcal{K}_k(A,r)$ tal que $Av = r$. Asimismo, como A es una matriz regular, $v = A^{-1}r \in \mathcal{K}_k(A,r)$.

(d) $\Rightarrow$ *(a)*: Sea $x = A^{-1}r \in \mathcal{K}_k(A,r)$ la solución del sistema de ecuaciones $Ax = r$. Entonces, $r \in \mathrm{span}\{Ar, \ldots, A^k r\}$ y $\{r, Ar, \ldots, A^k r\}$ son linealmente dependientes. □

Proposición 2.1.3. *Consideremos las hipótesis de la proposición 2.1.2. Sea* $\gamma = grado(A,r)$. *Entonces se cumple:*

i) $\mathcal{K}_k(A,r)$ *tiene dimensión* k *para todo* $k \leq \gamma$.

ii) $\mathcal{K}_k(A,r) = \mathcal{K}_\gamma(A,r)$ *para todo* $k > \gamma$.

Demostración. Demostremos ambos resultados por separado:

i) Si $k \leq \gamma$, entonces por la definición 2.1.2 no existe ningún polinomio $p(x)$ de grado menor o igual que $k-1$ tal que $p(A)r = 0$ y, por tanto, $r, Ar, \ldots, A^{k-1}r$ son linealmente independientes.

ii) La demostración utiliza la siguiente propiedad:

Sea $p(A) = \alpha_0 I + \alpha_1 A + \ldots + \alpha_{\gamma-1}A^{\gamma-1} + A^\gamma$ el polinomio mínimo de r respecto a A. Sea $q(x) = -(\alpha_0 I + \alpha_1 x + \ldots + \alpha_{\gamma-1}x^{\gamma-1})$. Notemos que $q(x)$ tiene grado $\gamma - 1$ y

$$q(A)r = A^\gamma r, \tag{2.1.5}$$

ya que $p(A)r = 0$.

Pasemos ahora a la demostración por inducción sobre k: si $k = \gamma + 1$, por (2.1.5) tenemos que $A^\gamma r$ es combinación lineal de $r, Ar, \ldots, A^{\gamma-1}r$ y, por tanto, $K_{\gamma+1}(A,r) = K_\gamma(A,r)$. Supongamos ahora que $\mathcal{K}_{\gamma+m}(A,r) = \mathcal{K}_\gamma(A,r)$ entonces

$$A^{\gamma+m}r = A^m A^\gamma r = A^m q(A)r \in \mathcal{K}_{\gamma+m}(A,r) = \mathcal{K}_\gamma(A,r),$$

donde se sigue que $\mathcal{K}_{\gamma+m+1}(A,r) = \mathcal{K}_\gamma(A,r)$.

□

Corolario 2.1.4. *En las condiciones de la proposición 2.1.2, la dimensión de* $\mathcal{K}_k(A,r)$ *es* k *si, y solo si,* $\gamma = grado(A,r) \geq k$.

Demostración. Primero probemos la implicación hacia la derecha. Por la segunda parte de la Proposición 2.1.3, si $\gamma < k$ entonces $\dim(\mathcal{K}_k(A, r)) = \gamma < k$. La otra implicación es consecuencia directa de la primera parte de la Proposición 2.1.3. □

El siguiente corolario es consecuencia directa de la Proposición 2.1.3.

Corolario 2.1.5. *En las condiciones de la proposición 2.1.2,* $\dim(\mathcal{K}_k(A, r)) = \text{mín}(k, \textit{grado}(A, r))$.

Una vez comprendido en que consisten estos particulares subespacios, adentrémonos en los llamados métodos de Krylov.

Consideremos el problema (2.1.1). Si denotamos la solución exacta como $x^* = A^{-1}b$ y $x_0 \in \mathbb{R}^n$ es una aproximación inicial a x^*, definimos entonces el vector residuo inicial como:

$$r_0 \coloneqq A(x^* - x_0) = b - Ax_0.$$

Asumimos que $r_0 \neq 0$. Entonces, resolver (2.1.1) equivale a resolver $Az = r_0$, con $z \coloneqq x - x_0$. Así, definimos también $z^* \coloneqq x^* - x_0$, de forma que $z^* = A^{-1}r_0$.

Los métodos de Krylov, como hemos mencionado antes, son métodos de proyección que se basan en proyectar nuestro problema (2.1.1) en un subespacio de Krylov. Concretamente, son una familia de métodos de proyección donde

$$\mathcal{K} = \mathcal{K}_k(A, r_0)$$

Existen varias razones para utilizar los métodos de Krylov en la resolución de sistemas de ecuaciones lineales, como la eficiencia computacional, pero sobre todo por la capacidad de manejar matrices de gran tamaño de manera efectiva. Son especialmente útiles cuando trabajamos con matrices de dimensión muy grande, las cuales contienen grandes bloques de ceros. Esto permite obtener soluciones de alta precisión con un costo computacional relativamente bajo en comparación con otros métodos. Aunque en este trabajo nos centramos en la utilidad de estos métodos para resolver (2.1.1), también son comúnmente usados para hallar los autovalores de una matriz cuadrada, [S]. Los métodos de Krylov incluyen varios algoritmos bien conocidos, de los cuales nos centraremos en dos:

- El método GMRES (Generalized Minimal Residual Method).
- El método de Arnoldi o FOM (Full Orthogonalization Method).

Estos son de gran interés en este trabajo puesto que, como veremos más adelante, son equivalentes en cierto sentido en el caso lineal al método de la aceleración de Anderson. FOM es apropiado para resolver sistemas lineales con matrices simétricas o bien acondicionadas. Está basado en el algoritmo de Arnoldi y es un método de proyección ortogonal, es decir, se tiene que $\mathcal{L} = \mathcal{K}$. En cambio, GMRES es eficaz cuando se trata de sistemas no simétricos y no definidos positivos. Es una buena elección cuando nuestra matriz está mal acondicionada. A diferencia del método de Arnoldi, en GMRES se tiene que $\mathcal{L} = A\mathcal{K}$.

Sin embargo, estos métodos presentan frecuentemente varios inconvenientes:

- Convergencia lenta: el mayor problema en ambos métodos es que en muchos casos la convergencia es lenta, especialmente cuando las matrices están mal acondicionadas o cuando los valores propios están distribuidos desfavorablemente (por ejemplo, cuando están dispersos o cerca del origen). En tales casos, el número de iteraciones requerido para obtener una solución precisa puede ser muy grande.

- Necesidad de preacondicionamiento:Para mejorar la convergencia, a menudo es necesario preacondicionar la matriz. El preacondicionamiento tiene como objetivo reducir el número de iteraciones necesarias, pero elegir un preacondicionador adecuado suele ser difícil, y su construcción puede ser costosa (véase el capítulo 9 de [S]).
- Costo computacional: ambos métodos son muy costosos cuando k es grande debido al crecimiento de los requisitos de memoria y computación a medida que k aumenta. Para reducir el esfuerzo computacional, son posibles implementaciones alternativas, como las basadas en el reinicio y el truncamiento, que describiremos más adelante.
- Sensibilidad a errores de redondeo: Ambos métodos utilizan bases ortogonales que, cuando se calculan en aritmética de punto flotante, pueden perder ortogonalidad debido a errores de redondeo.

2.1.3. Método de Arnoldi

El procedimiento fue introducido por primera vez por Walter Arnoldi en 1951 como un medio para reducir una matriz densa a la forma Hessenberg mediante una transformación unitaria, [AR]. Definamos este tipo de matrices, ya que hablaremos de ellas a lo largo de la subsección.

Definición 2.1.3. *Una matriz de Hessenberg superior (resp. inferior) es una matriz en la que todos los elementos por debajo de la subdiagonal (resp. por encima de la superdiagonal) son cero.*

Más tarde se descubrió que esta estrategia que usó Arnoldi, conduce a una técnica eficiente para aproximar los valores propios de matrices grandes y dispersas, y la técnica se extendió posteriormente a la resolución de sistemas lineales grandes y dispersos. Este método se basa en el llamado algoritmo de Arnoldi, el cual construye una base ortonormal de nuestro subespacio de Krylov, lo que lo hace particularmente útil cuando se trata de matrices dispersas grandes. El algoritmo de iteración de Arnoldi no solo encuentra una base ortonormal de $\mathcal{K}_k(A, r)$, sino que además calcula una matriz Hessenberg superior, que denotaremos por H_k. Esta jugará un papel importante en el algoritmo GMRES, el cual también hace uso del algoritmo de Arnoldi.

Observación 2.1.6. *De momento supondremos que* $\dim\left\{r, Ar, A^2r, \ldots, A^{k-1}r\right\} = k$*, descartando el caso grado*$(A, r) < k$*. El caso en que la dimensión de* $\mathcal{K}_k(A, r)$ *es menor que* k *lo trataremos en la subsección 2.1.4 en relación con GMRES.*

Algoritmo de Arnoldi

El algoritmo consiste en usar el proceso modificado de ortogonalización por Gram-Schmidt a partir de nuestra base $\left\{r, Ar, A^2r, \ldots, A^{k-1}r\right\}$, para llegar a una ortonormal $\{q_1, q_2, \ldots, q_{k+1}\}$.

Los q_j calculados por el algoritmo de Arnoldi se llaman vectores de Arnoldi. El proceso de obtención de los vectores $\widetilde{q_j}$ puede describirse a partir del algoritmo de Gram-Schmidt modificado. Nuestro primer elemento lo obtenemos simplemente normalizando r:

$$q_1 = \frac{r}{\|r\|_2}$$

A partir de aquí realizamos el siguiente algoritmo para $1 \leq i \leq k$ para obtener

$$\widetilde{q_{i+1}} = Aq_i - \sum_{j=1}^{i} q_j h_{ji}, \tag{2.1.6}$$

donde h_{ij} es el coeficiente de Gram-Schmidt

$$h_{ji} = \langle q_j, Aq_i \rangle = q_j^T A q_i,\ j \leq i,$$

con $\langle \cdot, \cdot \rangle$ el producto escalar euclídeo. En cada iteración, basta normalizar $\widetilde{q_{i+1}}$ para obtener los vectores ortonormales buscados.

$$q_{i+1} = \frac{\widetilde{q_{i+1}}}{\|\widetilde{q_{i+1}}\|_2}. \tag{2.1.7}$$

Los coeficientes de Gram-Schmidt se computan en este orden:

$$\begin{array}{cccc} h_{11} & h_{21} & & \\ h_{12} & h_{22} & h_{32} & \\ h_{13} & h_{23} & h_{33} & h_{34} \\ \vdots & & & \\ h_{1k} & h_{2k} & \cdots & h_{k+1,k} \end{array}$$

forman la matriz $(k+1) \times k$ de Hessenberg mencionada anteriormente:

$$\widetilde{H}_k = \begin{pmatrix} h_{11} & h_{21} & \cdots & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2k} \\ 0 & h_{32} & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & h_{k-1,k-1} & \vdots \\ \vdots & \vdots & \ddots & h_{k,k-1} & h_{k,k} \\ 0 & 0 & \cdots & 0 & h_{k+1,k} \end{pmatrix} \tag{2.1.8}$$

Observación 2.1.7. *Definimos la matriz $Q_k \in \mathbb{R}^{n \times k}$ como la que tiene por columnas los vectores $q_1, \ldots, q_k$. Si construimos un algoritmo iterativo que en la iteración $(k+1)$-ésima amplíe la matriz Q_k obtenida en la iteración k-ésima de forma que $Q_{k+1} = \left[Q_k \mid q_{k+1}\right]$, entonces tendremos un algoritmo que en cada iteración amplía una base ortonormal $q_1, \ldots, q_k$ de $\mathcal{K}_k(A, r)$ a una base ortonormal $q_1, \ldots, q_k, q_{k+1}$ de $\mathcal{K}_{k+1}(A, r)$.*

Podemos pensar en las matrices Q_k como en una única matriz Q que en cada iteración se amplía con una nueva columna.

Dado el subespacio de Krylov $\mathcal{K}_k(A, r)$, el algoritmo se implementaría de la siguiente manera:

Algoritmo 2: Iteración de Arnoldi

Entrada

- $A \in \mathbb{R}^{n\times n}$*: Matriz generadora de* $\mathcal{K}_k(A,r)$*.*
- $r \in \mathbb{R}^n$*: Vector inicial de* $\mathcal{K}_k(A,r)$*.*
- $k \in \mathbb{Z}$*: Dimensión de* $\mathcal{K}_k(A,r)$*.*

Salida

- $Q_{k+1} = \left(q_{i,j}\right) \in \mathbb{R}^{n\times(k+1)}$*: Matriz que tendrá por columnas los* $q_1, \ldots, q_{k+1}$*.*
- $\widetilde{H}_k = \left(h_{i,j}\right) \in \mathbb{R}^{(k+1)\times k}$*: Matriz descrita en* (2.1.8)*.*

Función `Arnoldi(`A, r, k`):`

$q_1 \leftarrow \frac{r}{\|r\|_2}$;
para $i \leftarrow 1$ **hasta** k **hacer**
  $t \leftarrow Aq_i$;
  para $j \leftarrow 1$ **hasta** i **hacer**
    $h_{j,i} \leftarrow q_j^T t$;
    $t \leftarrow t - h_{j,i}q_j$;
  $h_{i+1,i} \leftarrow \|t\|_2$;
  si $h_{i+1,i} \neq 0$ **entonces**
    $q_{i+1} \leftarrow \frac{t}{h_{i+1,i}}$;
devolver $[(q_{i,j})_{1\leq i\leq n,\, 1\leq j\leq k+1},\ (h_{i,j})_{1\leq i\leq k+1,\, 1\leq j\leq k}]$

Ejemplo 2.1. *Apliquemos el algoritmo anterior al siguiente ejemplo. Sean la matriz* A*, vector* r *y entero* k *siguientes:*

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}, \quad r = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad k = 3$$

Después de usar el algoritmo obtendríamos la matriz de de vectores ortogonales Q_3 *y la matriz* 4×3 *de Hessemberg* $\widetilde{H}_3$ *definida previamente en* (2.1.8)*:*

$$Q_4 = \begin{pmatrix} 0{,}5 & -0{,}6708 & 0{,}1693 & 0{,}3533 \\ 0{,}5 & -0{,}2236 & -0{,}0462 & -0{,}3006 \\ 0{,}5 & 0{,}2236 & -0{,}8774 & 0{,}4670 \\ 0{,}5 & 0{,}6708 & 0{,}4464 & 0{,}7528 \end{pmatrix}, \quad \widetilde{H}_3 = \begin{pmatrix} 34 & 4{,}4721 & -5{,}2339 \\ 17{,}8885 & 0 & -2{,}7537 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Es decir, la base ortonormalizada de $\mathcal{K}_3(A,r)$ *es*

$$\left\{ \begin{pmatrix} 0{,}5 \\ 0{,}5 \\ 0{,}5 \\ 0{,}5 \end{pmatrix}, \begin{pmatrix} -0{,}6708 \\ -0{,}2236 \\ 0{,}2236 \\ 0{,}6708 \end{pmatrix}, \begin{pmatrix} 0{,}1693 \\ -0{,}0462 \\ -0{,}8774 \\ 0{,}4464 \end{pmatrix} \right\}.$$

Este algoritmo tiene algunas variantes, como el algoritmo de Householder-Arnoldi, que usa reflectores de Householder para la ortogonalización, en lugar de la técnica clásica de Gram-Schmidt. Estos reflectores se utilizan para transformar una matriz a una forma más simple, generalmente una matriz Hessenberg, al aplicar transformaciones ortogonales que eliminan elementos debajo de la diagonal principal mientras preservan la ortogonalidad de los vectores en el subespacio de Krylov. De igual manera, existe otra versión dónde se utilizan transformaciones de Givens con el mismo objetivo, [S].

Algoritmo de Arnoldi aplicado a la resolución de sistemas lineales (FOM)

El método de Arnoldi o FOM es el método iterativo de proyección para resolver (2.1.1) basado en el algoritmo de Arnoldi, a partir de una base ortonormal para $\mathcal{K}_k(A, r_0)$, con $r_0 = b - Ax_0$ y x_0 el iterante inicial. En la implementación del método usamos una parte de la matriz $\widetilde{H}_k$ descrita en (2.1.8). En concreto $H_k \in \mathbb{R}^{k\times k}$, que es la matriz $\widetilde{H}_k$ después de eliminar su última fila. El método se implementa de la siguiente manera:

Algoritmo 3: Full Orthogonalization Method (FOM)

Entrada

- $A \in \mathbb{R}^{n\times n}$*: Matriz de coeficientes.*
- $b \in \mathbb{R}^n$*: Vector de términos independientes.*
- $x_0 \in \mathbb{R}^n$*: Aproximación inicial a la solución.*
- $k \in \mathbb{Z}$*: Dimensión del subespacio de Krylov.*

Salida

- $x_k \in \mathbb{R}^n$*: Aproximación k-ésima a la solución.*

Función `FOM`(A, b, x_0, k)**:**

$r_0 \leftarrow b - Ax_0$;
$\beta \leftarrow \|r_0\|_2$;
$[Q_{k+1}, \widetilde{H}_k] \leftarrow$ `Arnoldi`(A, r_0, k) (Algoritmo 2);
$y_k \leftarrow H_k^{-1}(\beta e_1)$;
$x_k \leftarrow x_0 + Q_k y_k$;
devolver x_k ;

Observación 2.1.8. *El algoritmo 3 depende de un parámetro k que es la dimensión del subespacio de Krylov. En la práctica, es deseable seleccionar k de manera dinámica. Esto sería posible si la norma del residuo de x_k estuviera disponible sin tener que calcular x_k. Entonces, el algoritmo podría detenerse en el paso adecuado. En este sentido, se tiene el siguiente resultado, cf. [S].*

Proposición 2.1.9. *El vector residuo de la solución aproximada x_k calculada por el algoritmo FOM se puede expresar como*

$$b - Ax_k = -h_{k+1,k} e_k^T y_k q_{k+1}.$$

Por lo tanto,

$$\|b - Ax_k\|_2 = h_{k+1,k} |e_k^T y_k|. \tag{2.1.9}$$

De este modo, la fórmula (2.1.9) *puede incorporarse al algoritmo 3 para su control a través de un entero de parada, con x_0 como el último iterante calculado.*

Ejemplo 2.2. *Para cada valor de j entero positivo, se consideran las matriz de Toeplitz simétricas $A^{(j)} \in \mathbb{R}^{j\times j}$, generadas respectivamente por un vector $v^{(j)} \in \mathbb{R}^j$, donde $v^{(j)} = \big(1,2,3,\ldots,j\big)^T$. Específicamente, para cada j, la matriz $A^{(j)}$ se construye tomando como primera columna y primera fila el vector $v^{(j)}$ extendido hasta completar las dimensiones requeridas. Tomemos para cada j, el sistema lineal*

$$A^{(j)}x^{(j)} = b^{(j)}, \tag{2.1.10}$$

dónde el vector de términos independientes $b^{(j)}$ será un vector de solo unos y el vector inicial $x_0^{(j)}$ será el vector nulo, ambos de tamaño j. Esto es:

$$A^{(j)} = \begin{pmatrix} 1 & 2 & 3 & \cdots & j-1 & j \\ 2 & 1 & 2 & \cdots & j-2 & j-1 \\ 3 & 2 & 1 & \cdots & j-3 & j-2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ j-1 & j-2 & j-3 & \cdots & 1 & 2 \\ j & j-1 & j-2 & \cdots & 2 & 1 \end{pmatrix}, \quad b^{(j)} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \quad x_0^{(j)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Tras aplicar FOM (algoritmo 3) para distintos valores de j y k, los residuos finales obtenidos se muestran en la tabla 2.1.

		k			
		10	20	30	40
j	100	$6{,}14 \times 10^{-3}$	$5{,}36 \times 10^{-5}$	$8{,}18 \times 10^{-8}$	$8{,}63 \times 10^{-12}$
	300	$1{,}67 \times 10^{-2}$	$1{,}92 \times 10^{-3}$	$2{,}21 \times 10^{-4}$	$1{,}30 \times 10^{-5}$
	500	$2{,}16 \times 10^{-2}$	$3{,}58 \times 10^{-3}$	$1{,}02 \times 10^{-3}$	$1{,}74 \times 10^{-4}$
	700	$6{,}11 \times 10^{-2}$	$4{,}74 \times 10^{-3}$	$1{,}55 \times 10^{-3}$	$5{,}76 \times 10^{-4}$
	900	$3{,}00 \times 10^{-1}$	$5{,}63 \times 10^{-3}$	$3{,}50 \times 10^{-3}$	$7{,}30 \times 10^{-4}$
	1100	$3{,}40 \times 10^{-2}$	$6{,}37 \times 10^{-3}$	$3{,}98 \times 10^{-3}$	$9{,}55 \times 10^{-4}$
	1300	$1{,}47 \times 10^{-1}$	$7{,}02 \times 10^{-3}$	$3{,}11 \times 10^{-3}$	$2{,}26 \times 10^{-3}$
	1500	$4{,}04 \times 10^{-2}$	$7{,}61 \times 10^{-3}$	$3{,}46 \times 10^{-3}$	$1{,}71 \times 10^{-3}$

Tabla 2.1: Norma del residuo k-ésimo obtenido al resolver (2.1.10) con FOM para diferentes valores de j y k.

Se puede observar que FOM muestra una convergencia razonable para matrices de Toeplitz pequeñas cuando k es grande, logrando residuos bajos. Sin embargo, a medida que j aumenta, incluso con $k = 40$, los residuos son relativamente grandes. Esto se debe al mal acondicionamiento de las matrices de Toeplitz, lo que causa un aumento en el residuo final. Esto es un claro indicador de que FOM enfrenta dificultades al manejar sistemas mal acondicionados, sugiriendo la necesidad de más iteraciones o de preacondicionar previamente.

Otras variantes del método de Arnoldi son FOM con reinicio, IOM (Implicit Orthogonalization Method) y DIOM (Deflated Implicit Orthogonalization Method), cada una de las cuales adapta el método de Arnoldi para mejorar su eficiencia y precisión en la resolución de (2.1.1). FOM reiniciado introduce un mecanismo de reinicio que permite ejecutar el método en bloques, reiniciándolo periódicamente para mantener las matrices de tamaño manejable y mejorar

la convergencia. IOM optimiza el proceso de ortogonalización utilizando un enfoque implícito, reduciendo el costo computacional y el uso de memoria al evitar el almacenamiento explícito de grandes matrices de ortogonalización. DIOM complementa esto con técnicas de deflación, acelerando la convergencia, [S].

2.1.4. Método GMRES

El método GMRES (Generalized Minimum Residual) fue propuesto por Saad y Schultz en 1986 como un método de subespacios de Krylov para la resolución de sistemas con matriz no singular y algún grado de dispersión, [SS]. Se trata de un método de proyección que se basa en tomar $\mathcal{L} = A\mathcal{K}_k(A, r_0)$ como espacio de restricciones. La idea fundamental de este método es expresar x_k en términos de la base ortonormal $\{q_1, \ldots, q_k\}$ generadora de $\mathcal{K}_k(A, r_0)$ usando el algoritmo de Arnoldi y en el cual q_1 es un múltiplo escalar del residuo inicial $r_0 = b - Ax_0$.

El método GMRES consiste en generar una sucesión de aproximaciones iterativas x_k que se mejoran en cada paso mediante un proceso de mínimos residuales. Veamos ahora una relación la cual necesitaremos para la construcción del algoritmo para GMRES.

Proposición 2.1.10. *Las matrices Q_k, Q_{k+1} definidas en la observación 2.1.7 y $\widetilde{H}_k$, definida en* (2.1.8)*, satisfacen la siguiente relación:*

$$AQ_k = Q_{k+1}\widetilde{H}_k \tag{2.1.11}$$

Demostración. De las relaciones (2.1.6) y (2.1.7) se tiene

$$Aq_i = \sum_{j=1}^{i} q_j h_{ji} + \widetilde{q_{i+1}} = \sum_{j=1}^{i} q_j h_{ji} + h_{i+1,i} q_{i+1} = \sum_{j=1}^{i+1} q_j h_{ji}. \tag{2.1.12}$$

Formamos la matriz $Q_k = \begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix} \in \mathbb{R}^{n\times k}$ y la matriz $Q_{k+1} = \begin{bmatrix} q_1 & q_2 & \cdots & q_k & q_{k+1} \end{bmatrix} \in \mathbb{R}^{n\times(k+1)}$ a partir de los vectores columna $\{q_i\}$, tal como se muestra en la observación 2.1.7. La ecuación matricial (2.1.11) se sigue directamente de la ecuación (2.1.12) □

Las soluciones que buscamos son de la forma

$$x_k = x_0 + Q_k y_k, \quad y_k \in \mathbb{R}^k, \tag{2.1.13}$$

donde Q_k es la matriz que tiene por columnas los vectores de $\mathbb{R}^n$ que forman una base ortogonal del subespacio de Krylov $\mathcal{K}_k(A, r_0) = \left\{r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0\right\}$ y donde el vector y_k es el que hace mínimo la norma del residuo k-ésimo. Es decir, en el cual

$$\|r_k\|_2 = \|b - A(x_0 + Q_k y_k)\|_2 = \|r_0 - AQ_k y_k\|_2$$

es mínimo en $\mathcal{K}_k(A, r_0)$. Este es un problema de mínimos cuadrados. A partir del algoritmo de Arnoldi, podemos simplificar el problema como sigue. Sea $\beta \coloneqq \|r_0\|_2$. Nuestra primera elección en el método de Arnoldi será $q_1 = \frac{r_0}{\|r_0\|_2}$, luego $r_0 = \beta q_1$. Por la relación (2.1.11) y puesto que $q_1 = Q_{k+1}\left(1, 0, \ldots, 0\right)^T = Q_{k+1}e_1$, se tiene que

$$\|r_k\|_2 = \|r_0 - AQ_k y_k\|_2 = \|\beta q_1 - Q_{k+1}\widetilde{H}_k y_k\|_2 = \|Q_{k+1}(\beta e_1 - \widetilde{H}_k y_k)\|_2.$$

Como las columnas de Q_{k+1} son ortonormales, basta con que minimicemos

$$\|\beta e_1 - \widetilde{H}_k y_k\|_2,$$

de manera que, resolviendo el sistema

$$\widetilde{H}_k y_k = \beta e_1, \tag{2.1.14}$$

en el sentido de mínimos cuadrados, obtenemos una solución aproximada de (2.1.1) en la forma (2.1.13). El problema (2.1.14) se resuelve aprovechando la estructura de la matriz H_k, donde es necesario descomponer la matriz en factorización QR con la ayuda de, o bien el método de Gram-Schmidt, o bien rotaciones de Givens oportunas, [F], o bien transformaciones de Householder, [S]. Una vez hallada la solución, este proceso se repite hasta que se alcanza una tolerancia deseada.

El algoritmo se implementa de la siguiente manera:

Algoritmo 4: GMRES

Entrada

- $A \in \mathbb{R}^{n\times n}$*: Matriz de coeficientes.*
- $b \in \mathbb{R}^n$*: Vector de términos independientes.*
- $x_0 \in \mathbb{R}^n$*: Aproximación inicial.*
- $k \in \mathbb{Z}$*: Dimensión de* $\mathcal{K}_k(A, r_0)$*, menor que* n*.*

Salida

- x_k*: Aproximación* k*-ésima a la solución.*

Función `GMRES`$(A,\ b,\ x_0,\ k)$**:**

$r_0 \leftarrow b - Ax_0$;
$\beta \leftarrow \|r_0\|_2$;
$[Q_{k+1}, \widetilde{H}_k] \leftarrow$ `Arnoldi`(A, r_0, k) (Algoritmo 2);
Determinar y_k que minimiza

$$\|\beta e_1 - \widetilde{H}_k y_k\|_2$$

$x_k \leftarrow x_0 + Q_k y_k$;
devolver x_k ;

Observación 2.1.11. *En la implementación práctica de GMRES, una estimación* x_k *a menudo no es suficiente para obtener la tolerancia de error deseada. Una opción viable es usar* x_k *como un vector inicial mejorado y repetir el proceso hasta alcanzar la tolerancia de error.*

El método GMRES converge a la solución exacta en exactamente n iteraciones, como veremos ahora en la proposición 2.1.13. Previamente, necesitamos la siguiente caracterización del residuo.

Proposición 2.1.12. *Sea* $A \in \mathbb{R}^{n\times n}$ *una matriz no singular y* x_k *la* k*-ésima iteración del GMRES, entonces para todo* $P \in \mathcal{P}_k = \{p\colon p \text{ es polinomio de grado } k \text{ y } p(0) = 1\}$*,*

$$\|r_k\|_2 = \min_{P\in\mathcal{P}_k} \|P(A)r_0\|_2.$$

Demostración. Sea $P \in \mathcal{P}_k$. Entonces podemos escribir P como $P(z) = 1 + p_1 z + \ldots + p_k z^k$, donde $1, p_1, \ldots, p_k$ son sus coeficientes. Consideramos la matriz V que tiene por columnas los vectores $\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$ que forman una base de $\mathcal{K}_k(A, r_0)$. Por tanto tendremos que

$$AV = A\begin{bmatrix} r_0 & Ar_0 & \cdots & A^{k-1}r_0 \end{bmatrix} = \begin{bmatrix} Ar_0 & A^2r_0 & \cdots & A^k r_0 \end{bmatrix}.$$

De esto se sigue que

$$\|r_k\|_2 = \|b - Ax_k\|_2 = \min_{y=(y_1,\ldots,y_k)^T} \|r_0 - AVy\| = \min_{y=(y_1,\ldots,y_k)^T} \|r_0 - y_1 Ar_0 - \ldots - y_k A^k r_0\|$$
$$= \min_{p=(p_1,\ldots,p_k)^T} \|r_0 - p_1 Ar_0 - \ldots - p_k A^k r_0\| = \min_{P\in\mathcal{P}_k} \|P(A)r_0\|.$$

□

Proposición 2.1.13. *Sea $A \in \mathbb{R}^{n\times n}$ una matriz no singular, entonces GMRES obtiene la solución exacta en n pasos.*

Demostración. Sea $p(z) = \det(A - zI)$ el polinomio característico de A, tal que su grado sea n. Observemos que $p(0) = \det(A)$, que es no nulo puesto que A es singular por hipótesis. Consideremos $q_n(z) = \frac{p(z)}{p(0)} \in \mathcal{P}_n$. Así, por el teorema de Cayley-Hamilton, $p(A) = \det(A)q_n(A) = 0$, y por la proposición 2.1.12, $r_k = b - Ax_k = 0$, en otras palabras, x_k es solución del sistema. □

Ejemplo 2.3. *La proposición 2.1.13 puede ilustrase con el siguiente ejemplo. Sean A y b de la siguiente manera:*

$$A = \begin{pmatrix} 4 & -1 & 1 \\ -1 & 3 & -2 \\ 1 & -2 & 3 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

Tras aplicar el algoritmo GMRES, nuestra solución aproximada es:

$$x = \begin{pmatrix} 0{,}2222 \\ 2{,}4444 \\ 2{,}5556 \end{pmatrix}.$$

En este ejemplo hemos realizado un total de 4 iteraciones, con un error relativo nulo.

Ejemplo 2.4. *Apliquemos ahora GMRES utilizando los mismos datos del ejemplo 2.2. Los resultados obtenidos para diferentes valores de j y k se muestran en la tabla 2.2.*

		k			
		10	20	30	40
j	100	$2{,}72 \times 10^{-2}$	$8{,}65 \times 10^{-3}$	$8{,}79 \times 10^{-3}$	$8{,}83 \times 10^{-3}$
	300	$4{,}99 \times 10^{-2}$	$2{,}09 \times 10^{-2}$	$1{,}85 \times 10^{-2}$	$1{,}86 \times 10^{-2}$
	500	$6{,}49 \times 10^{-2}$	$2{,}44 \times 10^{-2}$	$2{,}43 \times 10^{-2}$	$2{,}43 \times 10^{-2}$
	700	$7{,}80 \times 10^{-2}$	$2{,}90 \times 10^{-2}$	$2{,}98 \times 10^{-2}$	$2{,}89 \times 10^{-2}$
	900	$2{,}11 \times 10^{-1}$	$3{,}31 \times 10^{-2}$	$3{,}28 \times 10^{-2}$	$3{,}28 \times 10^{-2}$
	1100	$9{,}67 \times 10^{-2}$	$3{,}65 \times 10^{-2}$	$3{,}64 \times 10^{-2}$	$3{,}64 \times 10^{-2}$
	1300	$1{,}11 \times 10^{-1}$	$3{,}96 \times 10^{-2}$	$3{,}96 \times 10^{-2}$	$3{,}95 \times 10^{-2}$
	1500	$1{,}13 \times 10^{-1}$	$4{,}25 \times 10^{-2}$	$4{,}26 \times 10^{-2}$	$4{,}25 \times 10^{-2}$

Tabla 2.2: Norma del residuo k-ésimo obtenido al resolver (2.1.10) con GMRES para diferentes valores de j y k.

Como se puede observar, al igual que con FOM en el ejemplo 2.2, GMRES también es sensible al mal acondicionamiento de las matrices de Toeplitz, reflejando en ambos casos la necesidad del preacondicionamiento.

Como hemos comentado antes, GMRES se vuelve poco útil cuando k es grande debido al crecimiento de los requisitos de memoria y computación a medida que k aumenta. Estos requisitos son idénticos a los de FOM, y al igual que con este algoritmo, hay dos variaciones de GMRES que buscan solucionar estos problemas. Una está basada en el reinicio y la otra en truncar la ortogonalización de Arnoldi. Describamos estas dos variaciones de GMRES, [S]:

- **GMRES con reinicio:** En esta variante, después de aplicar GMRES, se reinicia el proceso y se comienza de nuevo utilizando la aproximación obtenida como aproximación inicial. Esta idea es la comentada en la observación 2.1.11. Este proceso se realiza hasta alcanzar una tolerancia deseada. De esta manera conseguimos mejores soluciones aunque el proceso sea mas lento, puesto que estamos aplicando GMRES varias veces, cada una de ellas con una aproximación inicial más próxima a la solución.
- **GMRES truncado:** En esta versión, en lugar de ortogonalizar completamente cada vector al llamar al algoritmo de Arnoldi, se utiliza una versión truncada de este procedimiento, la cual simplifica los cálculos y reduce el costo computacional. Esto resulta en una aproximación menos precisa, pero más eficiente. Por esto, aunque GMRES estándar puede darnos una mejor aproximación a la solución debido a que aprovecha un subespacio completo, la versión truncada lo hace con menos recursos computacionales y de memoria.

Relación entre GMRES y FOM

Por último, veamos qué relación existe entre los métodos de Arnoldi y GMRES. Sea ρ_j^F (resp. ρ_j^G) la norma del residuo lograda en el paso j por el FOM (resp. GMRES).

Proposición 2.1.14. *([S]) Supongamos que se han tomado k pasos del proceso de Arnoldi y que H_k no es singular. Entonces*

$$\rho_k^F = \frac{\rho_k^G}{\sqrt{1 - \left(\frac{\rho_k^G}{\rho_{k-1}^G}\right)^2}},$$

y por lo tanto

$$\rho_k^G = \frac{1}{\sqrt{\sum_{i=0}^{k} \left(\frac{1}{\rho_i^F}\right)^2}}. \tag{2.1.15}$$

Observación 2.1.15. *La proposición 2.1.14 sugiere la cercanía de ambos métodos. Notemos primero que, a partir de* (2.1.15) *se tiene que*

$$\rho_k^G \leq \rho_i^F, \quad i = 0, \ldots, k. \tag{2.1.16}$$

Por otro lado, definamos $\rho_{k\downarrow}^F = \min\{\rho_i^F,\, i = 0, \ldots, k\}$. Entonces, se tiene que

$$\frac{1}{(\rho_k^G)^2} = \sum_{i=0}^{k} \frac{1}{(\rho_i^F)^2} \leq \frac{k+1}{(\rho_{k\downarrow}^F)^2}. \tag{2.1.17}$$

Por lo tanto, de (2.1.16) *y* (2.1.17) *llegamos a la conclusión siguiente:*

$$\rho_k^G \leq \rho_{k\downarrow}^F \leq \sqrt{k+1}\rho_k^G.$$

2.2. Equivalencias entre la aceleración de Anderson y GMRES:

El objetivo de esta sección es establecer una equivalencia entre la aceleración de Anderson sin truncamiento, es decir, con $m_k = k$ para cada k, y GMRES en un sistema lineal. Asumimos por tanto, que nuestra función de iteración g es de la forma $Ax + b$ para una matriz de coeficientes $A \in \mathbb{R}^{n\times n}$ y un vector de términos independientes $b \in \mathbb{R}^n$. La relación (1.0.2) se puede escribir como

$$0 \equiv f(x) \equiv Ax + b - x = (A - I)\, x + b.$$

Esto permite interpretar la aceleración de Anderson como un método iterativo para el sistema lineal

$$(I - A)x = b, \tag{2.2.1}$$

donde se supone que $I - A$ es no singular. Explícitamente, del algoritmo 1, los iterantes son de la forma

$$x_k = \sum_{i=0}^{m_k} \alpha_i^{(k)} g(x_{k-m_k+i}) = g(x_{\text{mín}}),$$

con

$$x_{\text{mín}} = \sum_{i=0}^{m_k} \alpha_i^{(k)} x_{k-m_k+i}\,, \quad \sum_{i=0}^{m_k} \alpha_i^{(k)} = 1,$$

y residuo $r_k = b - (I - A)x_k = x_k - g(x_k)$. Siguiendo el algoritmo 1, observamos que $x_{\text{mín}}$ proporciona el mínimo residuo en el subespacio generado por $x_{k-m_k}, \ldots, x_k$.

Antes de empezar, especifiquemos la notación que usaremos a lo largo del capítulo.

Notación. *Para cada k, definimos x_k^{GMRES} y r_k^{GMRES}, que denotan la k-ésima iteración y el residuo de GMRES, respectivamente, y $\mathcal{K}_k \equiv \left\{r_0^{GMRES}, (I-A)r_0^{GMRES}, \ldots, (I-A)^{k-1}r_0^{GMRES}\right\}$, que denota el k-ésimo subespacio de Krylov generado por $(I - A)$ y r_0^{GMRES}. Definimos también la k-ésima iteración de Anderson por x_k^{AA}.*

El principal resultado de esta sección muestra que, bajo ciertas hipótesis, la aceleración de Anderson y GMRES son esencialmente equivalentes. Esto significa que las iteraciones obtenidas por cualquiera de los dos métodos se pueden generar a partir de los del otro. Esta equivalencia se puede extender a las diferentes variantes de GMRES, [WN]. Pero antes de adentrarnos en el resultado, veamos un pequeño lema que necesitaremos para la demostración del teorema.

Lema 2.2.1. *Supongamos que GMRES se aplica a $(I - A)x = b$ con $(I - A)$ no singular. Si $||r_{j-1}^{GMRES}||_2 > ||r_j^{GMRES}||_2 > 0$ para algún $j > 0$, entonces $r_j^{GMRES} \notin \mathcal{K}_j$.*

Demostración. Por conveniencia, definimos $\mathcal{K}_0 \equiv \{0\}$. Para $\ell \geq 0$, denotamos r_ℓ^{GMRES} como r_ℓ y denotamos la proyección ortogonal sobre $(I - A)(\mathcal{K}_\ell)^\perp$ por π_ℓ. Dado que $(I - A)(\mathcal{K}_\ell)^\perp \subseteq (I - A)(\mathcal{K}_{\ell-1})^\perp$ para $\ell = 1, \ldots, j$, demostramos por inducción que se verifica que $r_j = \pi_j r_{j-1}$. Supongamos que $r_j \neq 0$ para algún $j > 0$. Si $r_j \in \mathcal{K}_j$, entonces $r_j \in \mathcal{K}_j \cap (I - A)(\mathcal{K}_j)^\perp \subseteq \mathcal{K}_j \cap (I-A)(\mathcal{K}_{j-1})^\perp$. Dado que $\mathcal{K}_j \cap (I-A)(\mathcal{K}_{j-1})^\perp$ es un subespacio unidimensional que contiene a r_{j-1}, tenemos que $r_j = \lambda r_{j-1}$ para algún escalar λ. Entonces $r_j = \pi_j r_j = \lambda \pi_j r_{j-1} = \lambda r_j$. Dado que $r_j \neq 0$, se sigue que $\lambda = 1$ y $r_j = r_{j-1}$ y, en consecuencia, que $||r_{j-1}^{\text{GMRES}}||_2 = ||r_j^{\text{GMRES}}||_2$. □

Teorema 2.2.2. *Consideremos el problema* (1.0.1) *con $g(x) = Ax + b$, donde $A \in \mathbb{R}^{n\times n}$ y $b \in \mathbb{R}^n$ y supongamos lo siguiente:*

i) No hay truncamiento en la aceleración de Anderson, es decir, $m_k = k$ para cada k.

ii) $(I - A)$ *es no singular.*

iii) Se aplica GMRES al sistema (2.2.1)*, con* $x_0^{GMRES} = x_0^{AA}$.

iv) Para algún $k \geq 1$, $r_{k-1}^{GMRES} \neq 0$.

v) Para este mismo k, $\|r_{j-1}^{GMRES}\|_2 > \|r_j^{GMRES}\|_2$ *para cada* j *tal que* $0 < j < k$.

Entonces

$$\sum_{i=0}^{k} \alpha_i^{(k)} x_i^{AA} = x_k^{GMRES} \quad y \quad x_{k+1}^{AA} = g(x_k^{GMRES}).$$

Demostración. Por la hipótesis *iii)*, denotamos por comodidad $x_0 = x_0^{\text{GMRES}} = x_0^{\text{AA}}$. Para $i = 1, \ldots, k$, definimos $z_i = x_i^{\text{AA}} - x_0$. Para demostrar el teorema, necesitamos probar los dos siguientes lemas, bajo las hipótesis *i)* a *v)*:

Lema 2.2.3. *Para* $i \leq j \leq k$*, si* $\{z_1, \ldots, z_j\}$ *es una base de* $\mathcal{K}_j$*, entonces*

$$\sum_{i=0}^{j} \alpha_i^{(j)} x_i^{AA} = x_j^{GMRES} \quad y \quad x_{j+1}^{AA} = g(x_j^{GMRES}).$$

Demostración. Partimos de que

$$f_0 = g(x_0) - x_0 = Ax_0 + b - x_0 = b - (I - A)x_0 = r_0^{\text{GMRES}}. \tag{2.2.2}$$

Además, para $i = 1, \ldots, k$, tenemos que, como $g(x) = Ax + b$, de la hipótesis *iii)*

$$\begin{aligned} f_i &= g(x_i^{\text{AA}}) - x_i^{\text{AA}} \\ &= g(x_0 + z_i) - (x_0 + z_i) \\ &= g(x_0) - x_0 + (A - I)z_i \\ &= r_0^{\text{GMRES}} - (I - A)z_i. \end{aligned} \tag{2.2.3}$$

Por tanto, por (2.2.2) y (2.2.3), para $i \leq j \leq k$ y cualquier $\alpha = \left(\alpha_0, \ldots, \alpha_j\right)^T$, llegamos a que

$$F_j\alpha = \sum_{i=0}^{j} \alpha_i f_i = \left(\sum_{i=0}^{j} \alpha_i\right) r_0^{\text{GMRES}} - (I - A)\left(\sum_{i=0}^{j} \alpha_i z_i\right). \tag{2.2.4}$$

Luego, por (2.2.4), podemos verificar fácilmente que si $\{z_1, \ldots, z_j\}$ es una base de $\mathcal{K}_j$, entonces $\alpha^{(j)} = \left(\alpha_0^{(j)}, \alpha_1^{(j)}, \ldots, \alpha_j^{(j)}\right)^T$, con $\alpha_0^{(j)} = 1 - \sum_{i=1}^{j} \alpha_i^{(j)}$, resuelve el problema de minimización:

$$\min_{\alpha} \|F_j\alpha\|_2, \quad \text{sujeto a} \quad \sum_{i=0}^{j} \alpha_i = 1, \tag{2.2.5}$$

si y solo si $\left(\alpha_1^{(j)}, \ldots, \alpha_j^{(j)}\right)^T$ resuelve el problema de minimización de GMRES:

$$\min_{\alpha} \left\| r_0^{\text{GMRES}} - (I - A)\left(\sum_{i=0}^{j} \alpha_i z_i\right) \right\|_2. \tag{2.2.6}$$

Como consecuencia, la solución $\alpha^{(j)} = \left(\alpha_0^{(j)}, \ldots, \alpha_j^{(j)}\right)^T$ de (2.2.5) satisface:

$$\sum_{i=0}^{j} \alpha_i^{(j)} x_i^{AA} = \alpha_0^{(j)} x_0 + \sum_{i=1}^{j} \alpha_i^{(j)} (x_0 + z_i) = \left(\sum_{i=0}^{j} \alpha_i^{(j)}\right) x_0 + \sum_{i=1}^{j} \alpha_i^{(j)} z_i = x_0 + \sum_{i=1}^{j} \alpha_i^{(j)} z_i = x_j^{\text{GMRES}}. \tag{2.2.7}$$

Por tanto,

$$x_{j+1}^{AA} = \sum_{i=0}^{j} \alpha_i^{(j)} g(x_i^{AA}) = g\left(\sum_{i=0}^{j} \alpha_i^{(j)} x_i^{AA}\right) = g(x_j^{\text{GMRES}}).$$

□

Lema 2.2.4. *Para $1 \leq j \leq k$, $\{z_1, \ldots, z_j\}$ es una base de $\mathcal{K}_j$.*

Demostración. Por inducción sobre j, sea $z_1 = x_1^{\text{AA}} - x_0 = g(x_0)x_0 = r_0^{\text{GMRES}}$. Si de la hipótesis *iv)*, $k = 1$, entonces $r_0^{\text{GMRES}} \neq 0$. Si $k > 1$, por la hipótesis *v)* se tiene que $||r_0^{\text{GMRES}}||_2 > ||r_{k-1}^{\text{GMRES}}||_2 > 0$. Por tanto, $\{z_1\}$ es una base de $\mathcal{K}_1$.

Supongamos ahora que $k > 1$ y, como hipótesis inductiva, que $\{z_1, \ldots, z_j\}$ es una base de $\mathcal{K}_j$ para algún j tal que $1 \leq j < k$. Por el lema 2.2.3 y (2.2.7), tenemos que

$$\begin{aligned} z_{j+1} = x_{j+1}^{AA} - x_0 &= g(x_j^{\text{GMRES}}) - x_0 = Ax_j^{\text{GMRES}} + b - x_0 \\ &= Ax_j^{\text{GMRES}} + b - x_j^{\text{GMRES}} + \sum_{i=1}^{j} \alpha_i^{(j)} z_i = r_j^{\text{GMRES}} + \sum_{i=1}^{j} \alpha_i^{(j)} z_i. \end{aligned}$$

Dado que $r_j^{\text{GMRES}} \in \mathcal{K}_{j+1}$ y $\sum_{i=1}^{j} \alpha_i^{(j)} z_i \in \mathcal{K}_j$, tenemos que $z_{j+1} \in \mathcal{K}_{j+1}$. Además, dado que por las hipótesis *iv)* y *v)* se tiene

$$\|r_{j-1}^{\text{GMRES}}\|_2 > \|r_j^{\text{GMRES}}\|_2 \geq \|r_{k-1}^{\text{GMRES}}\|_2 > 0,$$

entonces se sigue por el lema 2.2.1 que $r_j^{\text{GMRES}} \in \mathcal{K}_j$. En consecuencia, z_{j+1} no puede depender linealmente de $\{z_1, \ldots, z_j\}$, y concluimos que $\{z_1, \ldots, z_{j+1}\}$ es una base de $\mathcal{K}_{j+1}$. Esto completa la inducción y la demostración del lema 2.2.4. □

Demostración del teorema 2.2.2: es inmediato a partir de los lemas 2.2.3 y 2.2.4. □

Las condiciones del teorema 2.2.2 admiten la posibilidad de que el residuo en la iteración k-ésima sea cero o que GMRES deje de avanzar a partir de la k-ésima iteración. Estos escenarios son especialmente relevantes y suelen presentarse con cierta frecuencia. Vamos a profundizar en ellos.

2.2.1. Caso en el que el residuo en la iteración k-ésima es cero

Si se cumplen las hipótesis del teorema y $r_k^{\text{GMRES}} = 0$, entonces $x_k^{\text{GMRES}} = g(x_k^{\text{GMRES}})$, es decir, x_k^{GMRES} es un punto fijo de g. Luego, el teorema implica que $x_{k+1}^{\text{AA}} = g(x_k^{\text{GMRES}}) = x_k^{\text{GMRES}}$ también es un punto fijo de g, y una implementación práctica del algoritmo AA terminaría en el paso $k + 1$, alcanzando la solución de (1.0.1).

Podemos observar que la equivalencia de los problemas de mínimos cuadrados (2.2.5) y (2.2.6) implica que (2.2.5) tiene una solución única y, por lo tanto, x_{k+1}^{AA} está definido de manera única en este caso, aunque F_k sea de rango deficiente según la siguiente proposición.

Proposición 2.2.5. *Supongamos que se cumplen las hipótesis del teorema 2.2.2. Entonces, rango*$(F_k) \geq k$*, y rango*$(F_k) = k$ *si y solo si* $r_k^{GMRES} = 0$.

Demostración. A partir de (2.2.4) con $j = k$, tenemos que

$$F_k \alpha = 0$$

para $\alpha = \big(\alpha_0, \ldots, \alpha_k\big)^T$ si y solo si

$$(I - A) \sum_{i=1}^{k} \alpha_i z_i = \sum_{i=0}^{k} \alpha_i r_0^{\text{GMRES}}. \tag{2.2.8}$$

Dado que $\{z_1, \ldots, z_k\}$ es linealmente independiente y $(I - A)$ es no singular, se sigue de (2.2.8) que

$$\sum_{i=0}^{k} \alpha_i = 0$$

si y solo si $\alpha = 0$. En consecuencia, al considerar soluciones no triviales de $F_k \alpha = 0$, podemos asumir sin pérdida de generalidad que

$$\sum_{i=0}^{k} \alpha_i = 1.$$

Supongamos que $\alpha = \big(\alpha_0, \ldots, \alpha_k\big)^T$ y $\tilde{\alpha} = \big(\tilde{\alpha}_0, \ldots, \tilde{\alpha_k}\big)^T$ satisfacen $F_k \alpha = F_k \tilde{\alpha} = 0$ y

$$\sum_{i=0}^{k} \alpha_i = \sum_{i=0}^{k} \tilde{\alpha}_i = 1.$$

Entonces, a partir de (2.2.8), su diferencia $\alpha - \tilde{\alpha}$ satisface

$$(I - A) \sum_{i=1}^{k} (\alpha_i - \tilde{\alpha}_i) z_i = 0.$$

Como arriba, se sigue que $\alpha = \tilde{\alpha}$. Se concluye que la dimensión del espacio nulo de F_k es a lo sumo uno; por lo tanto, rango$(F_k) \geq k$.

Además, observemos que rango$(F_k) = k$ si y solo si existe un $\alpha^{(k)} = \left(\alpha_0^{(k)}, \ldots, \alpha_k^{(k)}\right)^T$ tal que $F_k \alpha^{(k)} = 0$ y $\sum_{i=0}^{k} \alpha_i^{(k)} = 1$. Por (2.2.8), se verifica que esto se cumple si y solo si

$$(I - A) \sum_{i=1}^{k} \alpha_i^{(k)} z_i = r_0^{\text{GMRES}},$$

lo cual se cumple si y solo si $r_k^{\text{GMRES}} = 0$.

□

2.2.2. Caso de estancamiento de GMRES

En ciertos casos, al aplicar el método, el mal acondicionamiento de la matriz, la imprecisión numérica o la propia estructura del problema hacen que llegue un momento en el que ya no es posible acercarse más a la solución de (2.1.1).

Definición 2.2.1. *Un método iterativo se dice que se estanca en la iteración k-ésima cuando, a partir de esa iteración, deja de avanzar hacia una mejor solución. En otras palabras, las iteraciones posteriores no logran reducir de manera significativa el error o el residuo, lo que sugiere que el método ha dejado de converger o que la mejora en la aproximación de la solución es insignificante.*

Proposición 2.2.6. *Supongamos que se cumplen las hipótesis del teorema 2.2.2 y que* $r_{k-1}^{GMRES} = r_k^{GMRES} \neq 0$. *Entonces* $x_{k+1}^{AA} = x_k^{AA}$.

Demostración. Se sigue de la proposición 2.2.5 que rango$(F_k) = k + 1$; en consecuencia, el problema (1.2.1) (con $m_k = k$) tiene una solución única. Luego, dado que $r_{k-1}^{\text{GMRES}} = r_k^{\text{GMRES}}$, la solución debe ser de la forma $\alpha_k = \left(\alpha_0^{(k-1)}, \ldots, \alpha_{k-1}^{(k-1)}, 0\right)^T$, lo que implica

$$x_{k+1}^{\text{AA}} = \sum_{i=0}^{k} \alpha_i^{(k-1)} g(x_i^{\text{AA}}) = x_k^{\text{AA}}.$$

□

Si se cumplen las suposiciones del teorema 2.2.2 y GMRES estanca en el paso k con $r_{k-1}^{\text{GMRES}} = r_k^{\text{GMRES}} \neq 0$, entonces la proposición 2.2.6 muestra que el Algoritmo AA sin truncamiento satisface $x_{k+1}^{\text{AA}} = x_k^{\text{AA}}$. Se sigue entonces que $f_k = f_{k+1}$, es decir, las dos columnas finales de F_{k+1} son las mismas, y el problema de mínimos cuadrados

$$\min_{\alpha=(\alpha_0,\ldots,\alpha_{m_k})^T} \|F_k\alpha\|_2 \quad \text{s.a.} \quad \sum_{i=0}^{m_k} \alpha_i = 1 \tag{2.2.9}$$

en el paso $(k + 1)$ es deficiente en rango. Por esto, se esperaría que el estancamiento cercano de GMRES en algún paso resulte en una mala condición del problema de mínimos cuadrados en el siguiente paso. Esto representa una debilidad numérica potencial de la aceleración de Anderson en relación con GMRES, que no se descompone cuando se estanca antes de que se encuentre la solución. Como resultado, la condición del problema de mínimos cuadrados es una consideración central en la implementación del Algoritmo AA.

2.2.3. Preacondicionamiento del sistema

Como ya comentamos previamente, uno de los mayores problemas de los métodos iterativos en general es la lenta convergencia en algunos casos. La solución a este problema suele ser preacondicionar nuestro sistema. Utilizamos esta técnica con la intención de resolver (2.1.1) de tal manera que los métodos iterativos puedan converger más rápidamente y con mayor fiabilidad (véase el capítulo 9 de [S]).

Para el resto de esta sección, cambiamos ligeramente la notación. Consideramos resolver numéricamente un sistema (2.1.1) usando una iteración estacionaria clásica basada en una

descomposición de operadores $A = M - N$, donde M, $N \in \mathbb{R}^{n\times n}$ y M es no singular. Con esta descomposición, el sistema (2.1.1) se reformula como la ecuación de punto fijo

$$x = g(x) \equiv M^{-1}Nx + M^{-1}b,$$

y la iteración estacionaria es una iteración de punto fijo con esta g.

El siguiente corolario del teorema 2.2.2 afirma que, con este g, la aceleración de Anderson sin truncamiento es esencialmente equivalente en el sentido del teorema 2.2.2 a GMRES aplicado al sistema precondicionado izquierdo de M^{-1}, comenzando con el mismo x_0.

La notación es la misma, excepto que ahora el residuo GMRES k-ésimo es ahora $r_k^{\text{GMRES}} \equiv M^{-1}b - M^{-1}Ax_k^{\text{GMRES}}$ para cada k.

Corolario 2.2.7. *Asumimos las siguientes hipótesis:*

- $A = M - N$, *donde* $A \in \mathbb{R}^{n\times n}$ *y* $M \in \mathbb{R}^{n\times n}$ *son no singulares.*
- $g(x) = M^{-1}Nx + M^{-1}b$ *para* $b \in \mathbb{R}^n$.
- *La aceleración de Anderson no está truncada, es decir,* $m_k = k$ *para cada* k.
- *GMRES se aplica al sistema precondicionado izquierdo* $M^{-1}Ax = M^{-1}b$ *con punto inicial* $x_0^{GMRES} = x_0^{AA}$.
- *Para algún* $k \geq 1$, $r_{k-1}^{GMRES} \neq 0$. *Además, para cada* j *tal que* $1 \leq j < k$, $||r_{j-1}^{GMRES}||_2 > ||r_j^{GMRES}||_2$.

Entonces

$$\sum_{i=0}^{k} \alpha_i^{(k)} x_i^{AA} = x_k^{GMRES} \quad y \quad x_{k+1}^{AA} = g(x_k^{GMRES}).$$

Demostración. Aplicamos el teorema 2.2.2 con A y b reemplazados por $M^{-1}N$ y $M^{-1}b$, respectivamente. □

2.3. Resultados de convergencia de la aceleración para problemas lineales

El objetivo de esta sección es ver cómo se comporta la convergencia de la aceleración de Anderson en el caso lineal, [TK]. Como solo nos referimos a la aceleración de Anderson, escribiremos a partir de ahora x_k^{AA} como x_k. Recordemos que el caso lineal es donde la función de iteración es $g(x) = Ax + b$, con $A \in \mathbb{R}^{n\times n}$ regular y $b \in \mathbb{R}^n$. Por tanto el residuo k-ésimo de la aceleración (Algoritmo 1) es $f_k = g(x_k) - x_k = b - (I - A)x_k$. En esta sección probaremos que la convergencia de la sucesión de residuos es q-lineal. Pero veamos primero qué significa este tipo de convergencia.

Definición 2.3.1. *Una sucesión* $\{x_k\}$ *que converge a un límite* x^* *se dice que converge q-linealmente si existe una constante* c, *con* $0 \leq c < 1$, *tal que para todo* $k \geq 0$, *se cumple:*

$$\|x_{k+1} - x^*\|_2 \leq c\|x_k - x^*\|_2.$$

Teorema 2.3.1. *Si* $\|A\|_2 = c < 1$, *entonces la iteración de Anderson converge a* $x^* = (I - A)^{-1} b$ *y los residuos convergen q-linealmente a cero con factor q igual a c.*

Demostración. Dado que $\sum_{j=0}^{m_k} \alpha_j = 1$, del algoritmo 1 se tiene

$$f_{k+1} = b - (I - A)x_{k+1} = \sum_{j=0}^{m_k} \alpha_j [b - (I - A)(b + Ax_{k-m_k+j})]$$
$$= \sum_{j=0}^{m_k} \alpha_j A[b - (I - A)x_{k-m_k+j}] = A \sum_{j=0}^{m_k} \alpha_j f_{k-m_k+j}.$$

Del hecho de que si $\{\alpha_j\}_{j=0}^{m_k}$ es la solución del problema de los mínimos cuadrados (1.2.1), entonces, por definición,

$$\left\| \sum_{j=0}^{m_k} \alpha_j f_{k-m_k+j} \right\|_2 \leq \|f_k\|_2,$$

Por lo tanto, podemos concluir con que

$$\|f_{k+1}\|_2 \leq c\|f_k\|_2.$$

□

2.4. Ejemplo numérico

En esta sección, nuestro objetivo es mostrar en un caso práctico la relación que hemos visto entre el algoritmo AA y GMRES para el caso lineal. Queremos implementar ambos métodos para un problema de la forma (1.0.1) y comparar su residuo en función del número de iteraciones máximas. Vamos a trabajar con el problema presentado en [L]. Expliquémoslo brevemente.

Consideremos el problema de valor inicial y contorno siguiente.

$$\begin{cases} U_t(x,t) = U_{xx}(x,t) + f(x), & x \in [0,1],\ t \in [0,T], \\ U(0,t) = U(1,t) = 0, & t \in [0,T], \\ U(x,0) = v(x), & x \in [0,1], \\ v(0) = v(1) = 0. \end{cases} \tag{2.4.1}$$

donde $T > 0$ y se nos dan las funciones f y v. Los subíndices indican con respecto a qué variable se toma la derivada. El problema (2.4.1) se puede resolver analíticamente, si bien la solución se expresa en forma de una serie infinita. Alternativamente, el uso de diferencias finitas permite obtener aproximaciones que convergen a la solución bajo ciertas condiciones. En la próxima subsección veremos que, al ser (2.4.1) un problema lineal, la formulación en diferencias finitas nos lleva a tener que resolver un sistema lineal en cada paso, cuya resolución nos permite comparar los métodos AA y GMRES.

2.4.1. El Método de Diferencias Finitas y Euler Implícito

Fijados $N, M \geq 1$ enteros, discretizamos el intervalo $[0,1] \times [0,T]$ con una red uniforme de nodos (x_i, t_ℓ), con $i = 0, \ldots, N$, $\ell = 0, \ldots, M$, $x_i = i\Delta x$ y $t_\ell = \ell \Delta t$, con longitudes de paso $\Delta x = \frac{1}{N}$, $\Delta t = \frac{T}{M}$. El método de diferencias finitas está basado en buscar aproximaciones $U_{i,\ell}$ a los valores $U(x_i, t_\ell)$ de la solución, a partir de la evaluación de la primera ecuación de (2.4.1) en los nodos (x_i, t_ℓ). Con la condición de contorno $U(0,t) = U(1,t) = 0$, podemos considerar solo $\{x_1, \ldots, x_{N-1}\}$.

Para obtener la ecuación en diferencias finitas que satisface $U_{i,\ell}$, aproximamos las derivadas de (2.4.1) mediante las fórmulas

$$\begin{aligned} U_t(x_i,t_\ell) &\approx \frac{U(x_i,t_\ell)-U(x_i,t_{\ell-1})}{\Delta t}, \quad 1\le i\le N-1, \quad 1\le \ell\le M,\\ U_{xx}(x_i,t_\ell) &\approx \frac{U(x_{i+1},t_\ell)-2U(x_i,t_\ell)+U(x_{i-1},t_\ell)}{\Delta x^2}, \quad 1\le i\le N-1, \quad 0\le \ell\le M. \end{aligned}$$

Esto nos lleva a las ecuaciones en diferencias

$$\frac{U_{i,\ell}-U_{i,\ell-1}}{\Delta t}=\frac{U_{i+1,\ell}-2U_{i,\ell}+U_{i-1,\ell}}{\Delta x^2}+f(x_i), \quad 1\le i\le N-1, \quad 1\le \ell\le M, \tag{2.4.2}$$

con $U_{i,0}=v(x_i)$. En (2.4.2) se incluyen las condiciones de contorno $U_{0,\ell}=U_{N,\ell}=0$. Esto nos permite plantear la formulación matricial

$$\mathbf{U}_{\ell+1}=\mathbf{U}_\ell+\Delta t(A\mathbf{U}_{\ell+1}+\mathbf{f}), \quad 0\le \ell\le M-1, \tag{2.4.3}$$

donde $\mathbf{f}=\big(f(x_1),\dots,f(x_{N-1})\big)^T$, $A\in\mathbb{R}^{(N-1)\times(N-1)}$ es de la forma

$$A=\frac{1}{\Delta x^2}\begin{pmatrix} -2 & 1 & 0 & \cdots & 0\\ 1 & -2 & 1 & \cdots & 0\\ 0 & 1 & -2 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & \cdots & 0 & 1 & -2 \end{pmatrix},$$

y $\mathbf{U}_\ell=\big(U_{1,\ell},\dots,U_{N-1,\ell}\big)^T$, siendo $U_{i,\ell}\approx U(x_i,t_\ell)$. La solución obtenida vía (2.4.3) converge a la solución de (2.4.1) cuando $\Delta t,\Delta x\to 0$ con $\frac{\Delta t}{\Delta x^2}\le\frac{1}{2}$, [MM]. Podemos observar que (2.4.3) es una iteración de punto fijo para cada $0\le\ell\le M-1$ y, al expandir y ordenar los términos, obtenemos

$$\mathbf{U}_{\ell+1}=(\Delta tA)\mathbf{U}_{\ell+1}+(\mathbf{U}_\ell+\Delta t\mathbf{f}). \tag{2.4.4}$$

Además,vamos a definir $L=\Delta tA$ y, para cada t_ℓ,

$$\begin{aligned} b_\ell &:= \mathbf{U}_\ell+\Delta t\mathbf{f},\\ g_\ell(u) &\equiv Lu+b_\ell. \end{aligned} \tag{2.4.5}$$

Entonces (2.4.4) se escribe en la forma (1.0.1)

$$\mathbf{U}_{\ell+1}=g_\ell(\mathbf{U}_{\ell+1}). \tag{2.4.6}$$

Alternativamente, podemos escribir (2.4.4) como

$$(I-L)\mathbf{U}_{\ell+1}=b_\ell. \tag{2.4.7}$$

En consecuencia, no solo podemos aplicar, para cada $0\le\ell\le M-1$, la aceleración de Anderson al problema (2.4.6), sino también GMRES al problema equivalente (2.4.7).

2.4.2. Implementación

Para implementar los métodos de aceleración de Anderson (AA) y GMRES se ha utilizado MATLAB. El objetivo principal de este ejemplo es analizar la relación entre el Algoritmo AA y GMRES en un problema común de iteración lineal derivado de (2.4.3). Para simplificar el análisis y enfocarnos en la comparación, consideraremos únicamente el primer paso del esquema iterativo, es decir, $\ell = 0$.

En este caso, el dato inicial en los nodos se define como

$$\mathbf{U}_0 = (v(x_1), \ldots, v(x_{N-1}))^T,$$

lo que nos permite calcular b_0 según (2.4.5). El esquema iterativo para este primer paso toma la forma:

$$\mathbf{U}_1^{(k+1)} = g_0(\mathbf{U}_1^{(k)}), \quad k = 0, 1, \ldots,$$

donde g_0 viene dado por (2.4.5) con $\ell = 0$. Para mayor simplicidad en la notación, redefinimos

$$\mathbf{U} \coloneqq \mathbf{U}_1 \quad \text{y} \quad g \coloneqq g_0.$$

Para cada iteración k nos interesa el cálculo de la norma de los residuos de ambos métodos, denotados como $||r_k^{\text{AA}}||_2$ y $||r_k^{\text{GMRES}}||_2$.

Para la implementación, (2.4.5) requiere, además, un tamaño de paso de tiempo Δt, que en nuestros experimentos será $\Delta t = 10^{-6}$ y varios Δx que especificaremos después. Una entrada adicional para el algoritmo de aceleración de Anderson es el parámetro de truncación m_k, pero para este ejemplo tomaremos $m_k = k$, es decir, aplicaremos AA sin truncación.

En el caso de la aceleración de Anderson, el algoritmo requiere resolver un problema de mínimos cuadrados con restricción. Para hallar $\alpha^{(k)}$ en la iteración k-ésima, usaremos la función `lsqlin`$(F_k, \mathbf{0}_{N-1}, [\ ], [\ ], p_{k+1}, 1)$, donde $\mathbf{0}_{N-1}$ es el vector columna nulo de tamaño $N-1$ (tamaño de la red dadas las condiciones iniciales) y p_{k+1} es un vector fila de unos de tamaño $k+1$. Esencialmente, se está resolviendo el siguiente problema equivalente a (1.2.1):

$$\alpha^{(k)} = \min_{\alpha} \|F_k\alpha \ - \mathbf{0}_{N-1}\|_2 \quad \text{sujeto a} \quad p_{k+1}\alpha = 1.$$

Cabe señalar que Fang y Saad [FS] proponen resolver un problema de mínimos cuadrados sin restricciones equivalente a (1.2.1), el cual será presentado en el siguiente capítulo.

El programa compara el residuo obtenido por ambos métodos para un máximo de k iteraciones, parámetro que iremos ajustando según se vaya alcanzando o no la precisión de la máquina cuando ejecutemos el código. En nuestro ejemplo iniciaremos con el vector nulo como aproximación inicial.

Dado que en cada iteración es necesario determinar $F_k = \begin{bmatrix} f_0 & \cdots & f_k \end{bmatrix}$, se almacena el cálculo de F_k con el fin de evitar redundancias y no ralentizar el programa, puesto que $F_{k+1} = \begin{bmatrix} F_k \mid f_{k+1} \end{bmatrix}$ para $k \geq 1$ y $m_k = k$.

Finalmente, se genera una gráfica para cada caso que muestra la comparación entre los dos métodos.

2.4.3. Resultados

Dado el problema (2.4.3) en el paso de tiempo $\ell = 0$, consideremos los dos pares diferentes de v y f:

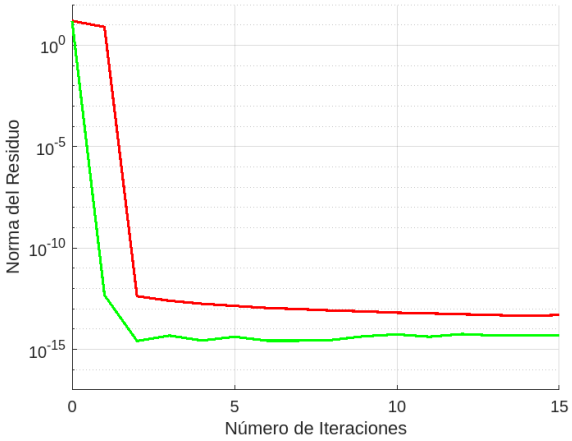
- **Caso 1:**

$$\begin{cases} f(x) \equiv \sin^2(x) \\ v(x) \equiv \sin(\pi x) \end{cases}$$

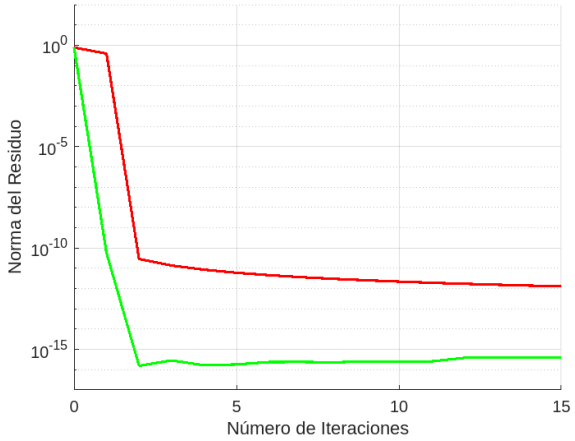
- **Caso 2:**

$$\begin{cases} f(x) \equiv \cos^2(x) \\ v(x) \equiv x^3 - \frac{3}{2}x^2 + \frac{1}{2}x \end{cases}$$

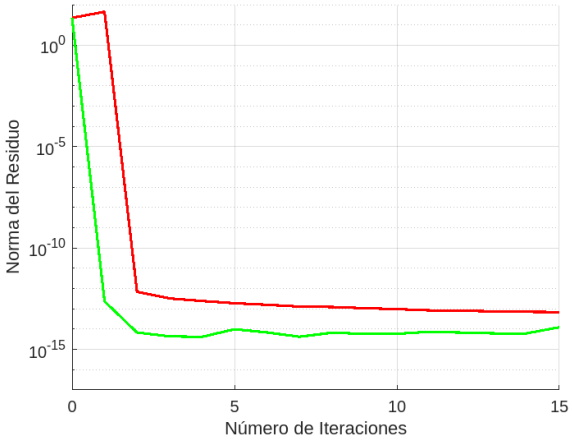
Es sencillo comprobar que en ambos casos v satisface las condiciones de compatibilidad $v(0) = v(1) = 0$. Probemos ambos casos para $N = 500$ y $N = 1000$, es decir, para $\Delta x = \frac{1}{500}$ y $\Delta x = \frac{1}{1000}$ respectivamente. Las gráficas de la norma relativa de los residuos en función del número de iteraciones para los métodos implementados en los 4 casos pueden verse en los gráficos 2.1a, 2.1b, 2.1c y 2.1d. Se ha usado $k = 15$ como número máximo de iteraciones, puesto que para valores más altos, la precisión de la máquina se alcanza y no se nos ofrece resultados interesantes para comparar.



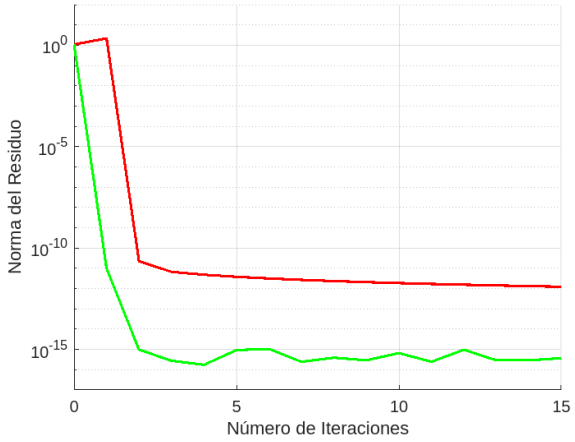
(a) Caso 1, $N = 500$.



(b) Caso 2, $N = 500$.



(c) Caso 1, $N = 1000$.



(d) Caso 2, $N = 1000$.

Gráfico 2.1: Comparación entre AA y GMRES para distintos tamaños de N y casos.

En vista de los resultados, podemos concluir que para estos ejemplos ambos métodos convergen rápidamente con velocidad similar. Podemos observar un pico en el caso de AA para $k = 1$. Esto se debe a que la primera aproximación está completamente determinada por $g(\mathbf{U}^{(0)})$, y a partir de este punto se comienza a minimizar el residuo.

Capítulo 3

El método AA en problemas no lineales

En este capítulo, nos centraremos en el problema (1.0.1) para el caso no lineal. Primero nos adentraremos en la familia a la que pertenece el método de aceleración de Anderson, los métodos multisecantes, concretamente multisecantes de Broyden. Veremos dónde se ubica el algoritmo AA (algoritmo 1) dentro de los métodos iterativos, conocido como Anderson "mixing", que es el término usado en literatura de estructuras electrónicas, [A, FS]. Después generalizaremos los métodos de Anderson como una familia, donde diferenciaremos dos variantes, tipo I y tipo II. En el capítulo 2 vimos cómo la aceleración de Anderson tiene cierta equivalencia al método GMRES cuando se trata de problemas lineales de la forma (2.1.1). En este capítulo, de manera análoga, mostraremos cómo en problemas lineales, un miembro particular de la familia de métodos de Anderson (el método de tipo I) guarda cierta equivalencia al método FOM (algoritmo 3).

3.1. Los métodos multisecantes

Consideremos el problema (1.0.1), el cual podemos reformular como $f(x) = 0$ a través de la relación (1.0.2). Para resolverlo, se emplean técnicas iterativas basadas en la construcción y mejora de aproximaciones sucesivas a la solución. La resolución mediante el enfoque iterativo puede dividirse en dos categorías principales:

- Métodos secuenciales que no requieren alta regularidad para f y que mejoran la convergencia de los vectores de aproximación en cada paso, [SI].
- Métodos derivados de la resolución numérica del sistema mediante aproximaciones del jacobiano $J_f(x) = \nabla f(x)$ o de su inversa. El ejemplo clásico en este sentido es el método de Newton, el cual presentamos en la introducción de este trabajo. Recordémoslo brevemente: si $f\colon \mathbb{R}^n \to \mathbb{R}^n$ es diferenciable con continuidad, a partir de la aproximación

$$f(x + \Delta x) \approx f(x) + J_f(x)\Delta x,$$

el método de Newton, en su iteración k-ésima, viene determinado por:

$$J_f(x_k)\Delta x_k = -f(x_k).$$

Entonces

$$x_{k+1} \coloneqq x_k + \Delta x_k = x_k - J_f(x_k)^{-1} f(x_k). \tag{3.1.1}$$

y $f(x_k + \Delta x_k) \approx 0$ si la solución Δx_k es suficientemente pequeña.

La elección del método iterativo adecuado está influenciada por una serie de limitaciones prácticas asociadas al cálculo y evaluación de f y sus derivadas:

1) Evaluación limitada de f debido a la alta dimensionalidad del sistema.

2) No disponibilidad explícita del jacobiano $J_f(x)$.

3) Alto costo computacional asociado a la evaluación de f y su jacobiano.

4) Uso limitado para evaluar f por errores que pueden presentar en su cálculo, [BC].

Estos desafíos motivan el diseño de métodos que minimicen el impacto de estas restricciones. En esta sección discutiremos este problema como sigue. Comenzaremos mencionando los métodos cuasi-Newton. Estos se basan en aproximaciones J_k al jacobiano $J_f(x_k)$ de forma eficiente, sin calcular derivadas de f. Una manera de hacer esto está dada por los métodos multisecantes. En ellos se busca J_k de manera que

$$f(x_k) + J_k(x - x_k) \tag{3.1.2}$$

interpole a f en x_k y x_{k+1}. Los métodos de este tipo más utilizados fueron propuestos por Broyden, [B].

La alternativa de usar aproximaciones de la inversa del jacobiano (en lugar del jacobiano mismo) lleva a la generalización de los métodos de Broyden (de los que el método de Anderson es un ejemplo) y Eirola-Nevalinna [EN].

Por último, estos métodos pueden modificarse actualizando la aproximación a la inversa del jacobiano a partir de estimaciones del jacobiano mismo. Presentaremos aquí dos familias de métodos, llamados de Broyden y de Anderson, [FS].

3.1.1. Métodos cuasi-Newton

En el método de Newton, el jacobiano se requiere en (3.1.1) para cada iteración, lo cual no es siempre práctico. Los métodos cuasi-Newton aproximan $J_f(x_k)$ con cierta matriz J_k, y obtienen J_{k+1} a partir de J_k añadiendo una matriz de bajo rango en cada iteración. Estos métodos aprovechan la información analítica sin calcular explícitamente el jacobiano real. Entre ellos, podemos destacar el método de Broyden, el método de Eirola-Nevalinna y Anderson mixing, [FS].

3.1.2. Método de Broyden

El método de Broyden se basa en calcular las aproximaciones J_k al jacobiano $J_f(x_k)$ imponiendo dos condiciones. Las mostramos a continuación, donde definimos $\Delta f_k \coloneqq f(x_{k+1}) - f(x_k)$ y $\Delta x_k = x_{k+1} - x_k$.

- Condición secante:
$$J_{k+1}\Delta x_k = \Delta f_k, \tag{3.1.3}$$
requerida por la condición mencionada anteriormente de que la aproximación lineal a $f(x)$ dada por (3.1.2) interpola a f en $x = x_{k+1}$.

- Condición "no change":
$$J_{k+1}q = J_k q \quad \forall q \text{ tal que } q^T \Delta x_k = 0, \tag{3.1.4}$$
que establece que pasar de J_k a J_{k+1} no cambia las direcciones ortogonales a Δx_k.

Broyden [B] desarrolló un método que satisface tanto la condición secante (3.1.3) como la condición "no change"(3.1.4). Esto se demuestra en el siguiente lema.

Lema 3.1.1. *Dadas las condiciones secante* (3.1.3) *y "no change"*(3.1.4)*, entonces se cumple*

$$J_{k+1} = J_k + (\Delta f_k - J_k \Delta x_k)\frac{\Delta x_k^T}{\Delta x_k^T \Delta x_k}. \tag{3.1.5}$$

Demostración. Dado $v \in \mathbb{R}^n$ arbitrario, lo descomponemos como combinación lineal de Δx_k y un elemento ortogonal. Esto es

$$v = \alpha q + \beta \Delta x_k, \qquad \text{con } q^T \Delta x_k = 0 \text{ y para ciertos } \alpha, \beta \in \mathbb{R}.$$

En particular,

$$\Delta x_k^T v = \beta \Delta x_k^T \Delta x_k \Rightarrow \beta = \frac{\Delta x_k^T v}{\Delta x_k^T \Delta x_k}.$$

Entonces, por (3.1.3) y (3.1.4), se tiene que

$$\begin{aligned} J_{k+1}v &= \alpha J_{k+1} q + \beta J_{k+1} \Delta x_k \\ &= \alpha J_k q + \beta \Delta f_k \\ &= J_k(\alpha q) + J_k(\beta \Delta x_k) + \beta(\Delta f_k - J_k \Delta x_k) \\ &= J_k v + (\Delta f_k - J_k \Delta x_k)\frac{\Delta x_k^T}{\Delta x_k^T \Delta x_k} v \\ &= \left(J_k + \frac{(\Delta f_k - J_k \Delta x_k)\Delta x_k^T}{\Delta x_k^T \Delta x_k} \right) v. \end{aligned}$$

□

La matriz J_{k+1} en (3.1.5) es la única que satisface ambas condiciones (3.1.3) y (3.1.4). Además, se puede demostrar (véase [DM]) que J_{k+1} se obtiene como solución del problema

$$\min_X \|X - J_k\|_F^2 \quad \text{sujeto a (3.1.3)},$$

donde $\|\cdot\|_F$ denota la norma de Frobenius. Puede parecer inicialmente que el método de Broyden es costoso, ya que calcular el paso cuasi-Newton Δx_k requiere resolver un sistema lineal en cada iteración. Sin embargo, el jacobiano aproximado es una modificación de bajo rango de una matriz diagonal, es decir, una matriz fácil de invertir. Por lo tanto el coste de pasar de J_k a J_{k+1} no es elevado si Δx_k es pequeño para un número de iteraciones moderado.

Una alternativa al método de Broyden, llamada método segundo de Broyden, aproxima la inversa del jacobiano en lugar del propio jacobiano, [FS]. Usamos G_k para denotar la aproximación del inverso del jacobiano en la iteración k-ésima. Podemos expresar las condiciones (3.1.3) y (3.1.4) de la forma

- Condición secante:

$$G_{k+1} \Delta f_k = \Delta x_k. \tag{3.1.6}$$

- Condición "no change":

$$G_{k+1} q = G_k q \quad \forall q \text{ tal que } q^T \Delta f_k = 0. \tag{3.1.7}$$

Igualmente, G_{k+1} se obtiene minimizando $E(X) = \|X - G_k\|_F$ sujeto a (3.1.6). Esto lleva a la fórmula de actualización

$$G_{k+1} = G_k + (\Delta x_k - G_k \Delta f_k)\frac{\Delta f_k^T}{\Delta f_k^T \Delta f_k},$$

la cual es la única que cumple las condiciones (3.1.6) y (3.1.7), [FS].

3.1.3. Método de Broyden generalizado y Anderson mixing

Veamos ahora una generalización del método de Broyden satisfaciendo un conjunto de m ecuaciones secantes, para $m \leq n$:

$$G_k \Delta f_i = \Delta x_i \quad \text{para } i = k-m, \ldots, k-1, \tag{3.1.8}$$

donde asumimos que $\Delta f_{k-m}, \ldots, \Delta f_{k-1}$ son linealmente independientes. Podemos reescribir (3.1.8) de forma matricial como

$$G_k \mathcal{F}_k = \mathcal{X}_k \tag{3.1.9}$$

donde

$$\mathcal{F}_k = \begin{bmatrix} \Delta f_{k-m} & \cdots & \Delta f_{k-1} \end{bmatrix} \quad \text{y} \quad \mathcal{X}_k = \begin{bmatrix} \Delta x_{k-m} & \cdots & \Delta x_{k-1} \end{bmatrix} \in \mathbb{R}^{n \times m}.$$

La condición "no-change"(3.1.7) en cada diferencia Δf_i con $i = k-m, \ldots, k-1$ da lugar a las ecuaciones

$$\begin{aligned} G_{k-m} q &= G_{k-m+1} q, \quad \forall q \text{ tal que } q^T \Delta f_{k-m} = 0, \\ G_{k-m+1} q &= G_{k-m+2} q, \quad \forall q \text{ tal que } q^T \Delta f_{k-m+1} = 0, \\ &\vdots \\ G_{k-1} q &= G_k q, \quad \forall q \text{ tal que } q^T \Delta f_{k-1} = 0, \end{aligned}$$

o, equivalentemente,

$$(G_k - G_{k-m})q = 0,$$

para todo q ortogonal al subespacio generado por $\Delta f_{k-m}, \ldots, \Delta f_{k-1}$, es decir, el espacio columna de $\mathcal{F}_k$, $\text{col}(\mathcal{F}_k)$. Entonces

$$\ker(G_k - G_{k-m}) = \text{span}\{\Delta f_{k-m}, \ldots, \Delta f_{k-1}\}^\perp = \text{col}(\mathcal{F}_k)^\perp.$$

Por tanto

$$\text{col}(\mathcal{F}_k) = ker(G_k - G_{k-m})^\perp = \text{col}((G_k - G_{k-m})^T).$$

Luego, para una cierta matriz desconocida Z,

$$(G_k - G_{k-m})^T = \mathcal{F}_k Z^T. \tag{3.1.10}$$

Trasponiendo (3.1.10) y multiplicando por $\mathcal{F}_k$ a la derecha, obtenemos

$$(G_k - G_{k-m})\mathcal{F}_k = Z\mathcal{F}_k^T \mathcal{F}_k \quad \Rightarrow \quad Z = (\mathcal{X}_k - G_{k-m}\mathcal{F}_k)(\mathcal{F}_k^T \mathcal{F}_k)^{-1},$$

donde hemos utilizado (3.1.9) y la hipótesis de que $\Delta f_{k-m}, \ldots, \Delta f_{k-1}$ son independientes, de manera que $\mathcal{F}_k^T \mathcal{F}_k$. Finalmente, esto produce

$$G_k = G_{k-m} + (\mathcal{X}_k - G_{k-m}\mathcal{F}_k)(\mathcal{F}_k^T \mathcal{F}_k)^{-1}\mathcal{F}_k^T, \tag{3.1.11}$$

que actualiza la aproximación a la inversa del jacobiano en el paso k a partir de la aproximación en el paso $k-m$. De nuevo, esta actualización de G_{k-m} en (3.1.11) se obtiene también a partir de la minimización de $E(X) = \|X - G_{k-m}\|_F$ sujeto a la condición (3.1.9).

La fórmula de actualización para x_{k+1} se deduce de (3.1.11) y (3.1.1):

$$\begin{aligned} x_{k+1} &= x_k - G_k f_k \\ &= x_k - G_{k-m} f_k - (\mathcal{X}_k - G_{k-m}\mathcal{F}_k)(\mathcal{F}_k^T \mathcal{F}_k)^{-1}\mathcal{F}_k^T \\ &= x_k - G_{k-m} f_k - (\mathcal{X}_k - G_{k-m}\mathcal{F}_k)\gamma_k, \end{aligned} \tag{3.1.12}$$

donde el vector columna γ_k se obtiene resolviendo las ecuaciones normales $(\mathcal{F}_k^T \mathcal{F}_k)\gamma_k = \mathcal{F}_k^T f_k$, que es equivalente a resolver el problema de mínimos cuadrados

$$\min_{\gamma} \|\mathcal{F}_k \gamma - f_k\|_2.$$

Este método es conocido como método segundo de Broyden generalizado.

Anderson mixing

Anderson Mixing, también conocido como aceleración de Anderson o simplemente método de Anderson (algoritmo 1), es equivalente al método segundo de Broyden generalizado como veremos a continuación.

Primero, reescribamos el problema de mínimos cuadrados (1.2.1). Tomemos $m = m_k$ y $\gamma = \left(\gamma_0, \ldots, \gamma_{m-1}\right)^T$ relacionado con $\alpha = \left(\alpha_0, \ldots, \alpha_m\right)^T$ de la siguiente manera:

$$\gamma_i = \sum_{j=0}^{i} \alpha_j, \quad \text{para } 0 \leq i \leq m-1. \tag{3.1.13}$$

Entonces (1.2.1) es equivalente al problema de mínimos cuadrados sin restricciones, [FS, A]

$$\min_{\gamma=(\gamma_0,\ldots,\gamma_{m-1})^T} \|f_k - \mathcal{F}_k \gamma\|_2, \tag{3.1.14}$$

Denotando la solución de (3.1.14) por $\gamma^{(k)} = \left(\gamma_0^{(k)}, \ldots, \gamma_{m-1}^{(k)}\right)^T$, podemos determinar $\gamma^{(k)}$ minimizando

$$E(x) = \|f_k - \mathcal{F}_k x\|_2^2,$$

a partir de las ecuaciones normales

$$\mathcal{F}_k^T \mathcal{F}_k \, x = \mathcal{F}_k^T f_k, \tag{3.1.15}$$

que tiene solución única $\gamma^{(k)}$ porque suponemos que rango$(\mathcal{F}_k) = m$. La ecuación de actualización del Algoritmo AA está dada por, [WN],

$$\begin{aligned} x_{k+1} &= g(x_k) - \sum_{i=0}^{m-1} \gamma_i^{(k)} \left(g\left(x_{k-m+i+1}\right) - g\left(x_{k-m+i}\right)\right) \\ &= x_k + f_k - \left(\mathcal{X}_k + \mathcal{F}_k\right)\gamma^{(k)}. \end{aligned} \tag{3.1.16}$$

Podemos por tanto implementar el algoritmo 1 de forma equivalente de la siguiente manera:

Algoritmo 5: Aceleración de Anderson sin restricciones

Entrada

- $x_0 \in \mathbb{R}^n$*: Solución aproximada inicial.*
- $g : \mathbb{R}^n \to \mathbb{R}^n$*: Función de iteración.*
- $m \in \mathbb{Z}$*: Truncamiento, que será mayor o igual que* 1*.*

Salida

- $x_{k+1} \in \mathbb{R}^n$*: Solución aproximada del sistema.*

Función `AA`(x_0, g, m)**:**

$x_1 \leftarrow g(x_0)$;
para $k \leftarrow 1, 2, \ldots$ **hacer**
$m_k \leftarrow \min(k, m)$;
$\mathcal{F}_k = \begin{bmatrix} \Delta f_{k-m_k} & \cdots & \Delta f_{k-1} \end{bmatrix}$, donde $\Delta f_j \coloneqq f_{j+1} - f_j$ y $f_j = g(x_j) - x_j$;
Determinar $\gamma^{(k)} = \left(\gamma_0^{(k)}, \ldots, \gamma_{m_k}^{(k)}\right)^T$ que resuelve

$$\min_{\gamma=(\gamma_0,\ldots,\gamma_{m-1})^T} \|f_k - \mathcal{F}_k\gamma\|_2,$$

$x_{k+1} = g(x_k) - \sum_{i=0}^{m_k-1} \gamma_j^{(k)} \Delta g_{k-m_k+i}$, donde $\Delta g_j \coloneqq g(x_{j+1}) - g(x_j)$;
devolver x_{k+1};

Sustituyendo la caracterización de la ecuación normal (3.1.15) en (3.1.16), se obtiene la forma cuasi-Newton

$$x_{k+1} = x_k + f_k - (\mathcal{X}_k + \mathcal{F}_k)\,(\mathcal{F}_k^T\mathcal{F}_k)^{-1}\mathcal{F}_k^T f_k. \tag{3.1.17}$$

Podemos observar que nuestra formula de actualización (3.1.17) es la misma que (3.1.12) tomando $G_{k-m} = -I$. En este contexto, el algoritmo AA es equivalente al método segundo de Broyden generalizado.

3.1.4. Familia de Broyden generalizada

La familia de Broyden generalizada es una variante del método segundo de Broyden generalizado de la sección 3.1.3, donde la actualización de G_k utiliza la aproximación al jacobiano $J_k \equiv G_k^{-1}$.

De manera análoga a la obtención de (3.1.11), obtenemos la fórmula de actualización para el jacobiano aproximado J_k

$$J_k = J_{k-m} + (\mathcal{F}_k - J_{k-m}\mathcal{X}_k)(\mathcal{X}_k^T\mathcal{X}_k)^{-1}\mathcal{X}_k^T, \tag{3.1.18}$$

a partir de los pasos siguientes:

- Satisfacer las condiciones secante $J_k\mathcal{X}_k = \mathcal{F}_k$ y "no change" $J_k q = J_{k-m} q$ para q ortogonal al subespacio generado por las columnas de $\mathcal{X}_k$.
- Minimizar $E(X) = \|X - J_{k-m}\|_F$ sujeto a $J_k\mathcal{X}_k = \mathcal{F}_k$.

La fórmula (3.1.18) se puede expresar en términos de $G_k \equiv J_k^{-1}$ aplicando la fórmula de Woodbury, [W]:

$$G_k = G_{k-m} + (\mathcal{X}_k - G_{k-m}\mathcal{F}_k)(\mathcal{X}_k^T G_{k-m}\mathcal{F}_k)^{-1}\mathcal{X}_k^T G_{k-m}. \tag{3.1.19}$$

En vista de las claras diferencias entre (3.1.11) y (3.1.19), pasemos a distinguir métodos de dos tipos, [FS].

Definición 3.1.1. *Dado un conjunto de ecuaciones secantes representado por $\mathcal{X}_k$ y $\mathcal{F}_k$, definimos:*

- *Métodos Tipo-I: aquellos métodos que actualizan G_k mediante* (3.1.19)*, minimizando el cambio del Jacobiano aproximado en la norma de Frobenius. Los métodos de Tipo-I se diferencian entre sí por cómo se selecciona el conjunto de ecuaciones secantes.*
- *Métodos Tipo-II: aquellos que actualizan G_k mediante la fórmula* (3.1.11)*, minimizando el cambio del Jacobiano inverso aproximado en la norma de Frobenius. Al igual que los métodos de Tipo-I, los métodos de Tipo-II también se diferencian entre sí por la forma en que se elige el conjunto de ecuaciones secantes.*

Ahora podemos escribir la familia generalizada de Broyden, en la que un algoritmo de actualización toma la forma

$$G_k = G_{k-m} + (\mathcal{X}_k - G_{k-m}\mathcal{F}_k)V_k^T,$$

donde $V_k^T\mathcal{F}_k = I$, de modo que se cumple la condición secante (3.1.9). El primer método generalizado de Broyden (ecuación (3.1.19)) y el segundo método generalizado de Broyden (ecuación (3.1.11)) son dos miembros particulares de esta familia, que ahora podemos llamar métodos de Broyden generalizado tipo I y tipo II respectivamente.

3.1.5. Familia de Anderson

Recordemos que el método de Anderson mixing forma implícitamente el Jacobiano inverso aproximado G_k y minimiza $\|G_k + I\|_F$, sujeto a las ecuaciones secantes disponibles. Como vimos, el método generalizado de Broyden presentado en la subsección 3.1.3, es un miembro particular de la familia generalizada de Broyden, y que podemos ahora llamar método generalizado de Broyden tipo II. También hemos visto que el método Anderson mixing presentado en la subsección 3.1.3, es el algoritmo AA (algoritmo 1) y, a su vez, es equivalente al método generalizado de Broyden tipo II en este contexto. El objetivo de esta sección es presentar la familia de Anderson, donde distinguiremos Anderson tipo I y II, miembros particulares de esta familia, de manera análoga a como lo hicimos con la familia generalizada de Broyden. Por último veremos que el algoritmo AA o Anderson mixing se puede identificar, mediante la definición 3.1.1, como el método de Anderson tipo II.

Para definir la familia de Anderson, generalizamos (3.1.17) reemplazando $(\mathcal{F}_k^T\mathcal{F}_k)^{-1}\mathcal{F}_k^T$ por una matriz V_k^T,

$$x_{k+1} = x_k + f_k - (\mathcal{X}_k + \mathcal{F}_k)V_k^T f_k, \tag{3.1.20}$$

donde $V_k \in \mathbb{R}^{n\times m}$ satisface $V_k^T\mathcal{F}_k = I$ para cumplir con la condición secante (3.1.9).

Puesto que el método de Anderson mixing corresponde al método de Broyden generalizado tipo II, lo podemos clasificar como un método de tipo II. Por lo tanto, a partir de ahora lo nombraremos como Anderson mixing tipo II.

De manera similar, tomando $V_k^T = (\mathcal{X}_k^T \mathcal{F}_k)^{-1} \mathcal{X}_k^T$ en (3.1.20), obtenemos así el método Anderson mixing tipo I

$$x_{k+1} = x_k + f_k - (\mathcal{X}_k + \mathcal{F}_k)(\mathcal{X}_k^T \mathcal{F}_k)^{-1} \mathcal{X}_k^T f_k, \tag{3.1.21}$$

que corresponde al jacobiano aproximado J_k en (3.1.18) con $J_{k-m} = -I$, por lo que satisface las condiciones secante $J_k \mathcal{X}_k = \mathcal{F}_k$ y minimiza $\|J_k + I\|_F$.

3.2. Equivalencias entre aceleración de Anderson tipo I y FOM

El objetivo de esta sección es establecer, en sistemas del tipo (2.2.1), cierta equivalencia entre FOM (algoritmo 3) y la aceleración de Anderson tipo I, es decir, el método que tiene a (3.1.21) como ecuación de actualización, [WN]. En esta sección suponemos entonces que g es de la forma (2.0.1).

Notación. *Para cada j, x_j^{FOM} y r_j^{FOM} denotan la j-ésima iteración y residuo del método FOM respectivamente. $\mathcal{K}_j$ denota el j-ésimo subespacio de Krylov generado por $(I - A)$ y r_0^{FOM}. Por último, $x_j^{Tipo\ I}$ denota la j-ésima iteración del método de Anderson tipo I.*

Teorema 3.2.1. *Consideremos el problema* (1.0.1) *con g de la forma* (2.0.1) *y supongamos lo siguiente:*

i) El método de Tipo I no está truncado, es decir, $m_k = k$ para cada k.

ii) $(I - A)$ es no singular.

iii) FOM se aplica al sistema (2.2.1) *con $x_0^{FOM} = x_0^{Tipo\ I}$.*

iv) Para algún $k \geq 1$, x_j^{FOM} está definido para $1 \leq j \leq k$ y $r_{k-1}^{FOM} \neq 0$.

Entonces $\mathcal{X}_j^T \mathcal{F}_j$ es no singular para $1 \leq j \leq k$ y, con $\alpha^{(k)} = \left(\alpha_0^{(k)}, \ldots, \alpha_{m_k}^{(k)}\right)^T$ dado por (3.1.13) *con $\alpha_{m_k}^{(k)} = 1 - \gamma_{m_k}^{(k)}$ y* (3.1.15)*. Además,*

$$\sum_{i=0}^{k} \alpha_i^{(k)} x_i^{Tipo\ I} = x_k^{FOM} \quad y \quad x_{k+1}^{Tipo\ I} = g(x_k^{FOM}).$$

Demostración. La demostración sigue un recorrido similar a la del teorema (2.2.2), salvo por algunos detalles. Por la hipótesis *iii)*, denotamos por comodidad $x_0 = x_0^{\text{FOM}} = x_0^{\text{Tipo I}}$. Para $i = 1, \ldots, k$, definimos $z_i = x_i^{\text{Tipo I}} - x_0$. Para demostrar el teorema, necesitamos probar los dos siguientes lemas, bajo las hipótesis *i)* a *iv)*:

Lema 3.2.2. *Para $i \leq j \leq k$, si $\{z_1, \ldots, z_j\}$ es una base de K_j, entonces*

$$\sum_{i=0}^{j} \alpha_i^{(j)} x_i^{Tipo\ I} = x_j^{FOM} \quad y \quad x_{j+1}^{Tipo\ I} = g(x_j^{FOM}).$$

Demostración. Para $i \leq j \leq k$ y cualquier $\gamma = \left(\gamma_0, \ldots, \gamma_{j-1}\right)^T$, se cumple:

$$f_j - \mathcal{F}_j\gamma = f_j - \sum_{i=0}^{j-1} \gamma_i(f_{i+1} - f_i) = \alpha_0 f_0 + \alpha_1 f_1 + \ldots + \alpha_{j-1} f_{j-1} + \alpha_j f_j, \tag{3.2.1}$$

donde $\alpha_0 = \gamma_0$, $\alpha_i = \gamma_i - \gamma_{i-1}$ para $1 \leq i < j$, y $\alpha_j = 1 - \gamma_{j-1}$. Por lo tanto,

$$\sum_{i=0}^{j} \alpha_i = \gamma_0 + (\gamma_1 - \gamma_0) + \ldots + (\gamma_j - \gamma_{j-1}) + (1 - \gamma_{j-1}) = 1. \tag{3.2.2}$$

Además, se tiene que

$$f_0 = g(x_0) - x_0 = Ax_0 + b - x_0 = b - (I - A)x_0 = r_0^{\text{FOM}}. \tag{3.2.3}$$

De manera similar, para $i = 1, \ldots, k$, tenemos que, como $g(x) = Ax + b$, de (3.2.3) y la hipótesis *iii)*:

$$\begin{aligned} f_i &= g(x_i^{\text{Tipo I}}) - x_i^{\text{Tipo I}} = g(x_0 + z_i) - (x_0 + z_i) \\ &= g(x_0) - x_0 + (A - I)z_i = r_0^{\text{FOM}} - (I - A)z_i. \end{aligned} \tag{3.2.4}$$

Entonces, por (3.2.1), (3.2.2) y (3.2.4),

$$f_j - \mathcal{F}_j\gamma = r_0^{\text{FOM}} - (I - A)\sum_{i=1}^{j} \alpha_i z_i,$$

lo que significa que $f_j - \mathcal{F}_j\gamma$ es el residuo asociado con $x_0 + \sum_{i=1}^{j} \alpha_i z_i$, que resulta del paso $\sum_{i=1}^{j} \alpha_i z_i \in \mathcal{K}_j$. Dado que

$$\mathcal{K}_j = \text{span}\{z_1, z_2, \ldots, z_j\} = \text{span}\{z_1, z_2 - z_1, \ldots, z_j - z_{j-1}\} = \text{span}\{\Delta x_0, \Delta x_1, \ldots, \Delta x_{j-1}\},$$

entonces la condición de residuo ortogonal que caracteriza de manera única x_j^{FOM} (como método de proyección), si está definido, es:

$$\mathcal{X}_j^T\left(f_j - \mathcal{F}_j\gamma\right) = \mathcal{X}_j^T\left[r_0^{\text{FOM}} - (I - A)\sum_{i=1}^{j} \alpha_i z_i\right] = 0. \tag{3.2.5}$$

Por lo tanto, nuestra suposición de que x_j^{FOM} está definido implica que existe un único γ (y su correspondiente α) que resuelven el sistema (3.2.5). De esto se deduce que $\mathcal{X}_j^T\mathcal{F}_j$ es no singular. Además, la solución es

$$\gamma^{(j)} = (\mathcal{X}_j\mathcal{F}_j)^{-1}\mathcal{X}_j^T f_j,$$

y el correspondiente $\alpha^{(j)}$ satisface, usando (3.2.2) y la definición de los z_i,

$$x_j^{\text{FOM}} = x_0 + \sum_{i=1}^{j} \alpha_i^{(j)} z_i = \sum_{i=0}^{j} \alpha_i^{(j)} x_i^{\text{Tipo I}},$$

de donde se sigue que

$$x_{j+1}^{\text{Tipo I}} = g(x_j^{\text{FOM}}). \tag{3.2.6}$$

□

Lema 3.2.3. *Para $1 \leq j \leq k$, $\{z_1, \ldots, z_j\}$ es una base de $\mathcal{K}_j$.*

Demostración. Por inducción sobre j. De la hipótesis *iii)* se tiene que $z_1 = g(x_0) - x_0 = r_0^{\text{FOM}}$. Por la hipótesis *iv)* llegamos a que $r_{k-1}^{\text{FOM}} \neq 0$. Por lo tanto, $\{z_1\}$ es una base de $\mathcal{K}_1$. Supongamos ahora que $\{z_1, \ldots, z_j\}$ es una base de $\mathcal{K}_j$ para algún j con $1 \leq j < k$. Por (3.2.6), se tiene que

$$\begin{aligned} z_{j+1} = x_{j+1}^{\text{Tipo I}} - x_0 = g(x_j^{\text{FOM}}) - x_0 = Ax_j^{\text{FOM}} + b - x_0 \\ = b - (I - A)x_j^{\text{FOM}} + x_j^{\text{FOM}} - x_0 = r_j^{\text{FOM}} + \sum_{i=1}^{j} \alpha_i^{(j)} z_i. \end{aligned}$$

Sabemos que $r_j^{\text{FOM}} \in \mathcal{K}_{j+1} \cap \mathcal{K}_j^{\perp}$ y $r_j^{\text{FOM}} \neq 0$ porque $r_{k-1}^{\text{FOM}} \neq 0$. Además, dado que $\sum_{i=1}^{j} \alpha_i^{(j)} z_i \in \mathcal{K}_j$, se deduce que z_{j+1} no puede depender linealmente de $\{z_1, \ldots, z_j\}$. Por lo tanto, $\{z_1, \ldots, z_{j+1}\}$ es una base de $\mathcal{K}_{j+1}$. Esto completa la inducción y la demostración del lema 3.2.3. □

Demostración del teorema 3.2.1: es inmediato a partir de los lemas 3.2.2 y 3.2.3. □

Concluimos esta sección con un análogo del corolario 2.2.7 para el caso en el que una iteración estacionaria basada en una descomposición del operador $A = M - N$ se aplica para resolver (2.1.1).

Corolario 3.2.4. *Asumamos las siguientes hipótesis:*

- *$A = M - N$, donde $A \in \mathbb{R}^{n\times n}$ y $M \in \mathbb{R}^{n\times n}$ son no singulares.*
- *$g(x) = M^{-1}Nx + M^{-1}b$, con $b \in \mathbb{R}^n$.*
- *El método de Tipo I no está truncado, es decir, $m_k = k$ para cada k.*
- *FOM se aplica al sistema precondicionado por la izquierda $M^{-1}Ax = M^{-1}b$ con $x_0^{FOM} = x_0^{Tipo\ I}$.*
- *Para algún $k \geq 1$, x_j^{FOM} está definido para $1 \leq j \leq k$ y $r_{k-1}^{FOM} \neq 0$.*

Entonces, con $\alpha^{(k)} = \left(\alpha_0^{(k)}, \ldots, \alpha_{m_k}^{(k)}\right)^T$ dado por (3.1.13) *y* (3.1.15),

$$\sum_{i=0}^{k} \alpha_i^{(k)} x_i^{Tipo\ I} = x_k^{FOM} \quad y \quad x_{k+1}^{Tipo\ I} = g(x_k^{FOM}).$$

Demostración. Aplicamos el teorema 3.2.1 reemplazando A y b por $M^{-1}N$ y $M^{-1}b$, respectivamente. □

3.3. Resultados de convergencia de la aceleración para problemas no necesariamente lineales

En esta sección mencionamos brevemente algunos resultados de convergencia del método AA en el caso no lineal, cuya demostración puede seguirse en [TK]. Hay que tener en cuenta que, por un lado, no hay resultados generales (no hay una teoría de convergencia para los métodos multisecantes) y, por otro, que la mayoría de las aplicaciones utilizan valores pequeños de m, en comparación con el tamaño n del sistema.

De los resultados obtenidos en [TK], destacamos el referido al caso $m = 1$ con g regular y contractiva en una bola en $\mathbb{R}^n$. Entonces los residuos del método convergen q-linealmente (véase la definición 2.3.1).

Teorema 3.3.1. *([TK]) Supongamos que se cumplen las siguientes hipótesis:*

i) Existe $x^* \in \mathbb{R}^n$ *tal que* $f^* = g(x^*) - x^* = 0$.

ii) g *es continuamente diferenciable en una bola* $B(\rho) = \{x \colon \|x\| \leq \rho\}$ *para algún* $\rho > 0$.

iii) La función g *es contractiva en* $B(\rho)$*, es decir, existe* $c \in (0,1)$ *tal que*

$$\|g(u) - g(v)\|_2 \leq c\|u - v\|_2 \quad \text{para todos } u, v \in B(\rho).$$

iv) $x_0 \in B(\rho)$.

v) c *es lo suficientemente pequeño como para que*

$$\hat{c} \equiv \frac{3c - c^2}{1 - c} < 1. \tag{3.3.1}$$

Entonces, los residuos de Anderson con $m = 1$ *convergen q-linealmente con factor* q *igual a* $\hat{c}$.

Observación 3.3.2. *El resultado anterior puede generalizarse a* $m > 1$ *y si el problema se plantea en diferentes normas, bajo la hipótesis adicional de que en cada paso* k *los coeficientes* $\alpha_j^{(k)}$ *están acotados, e. g. [TK] para detalles.*

3.4. Ejemplos numéricos

En esta sección, presentamos tres ejemplos para ilustrar los resultados previos de la aceleración de Anderson, uno lineal, para completar lo desarrollado en el capítulo 2, y otros dos referidos a problemas no lineales.

3.4.1. Primer ejemplo: Problema PageRank

El algoritmo PageRank, desarrollado por Google, se utiliza para clasificar páginas web en función de su importancia relativa dentro de una red de hiperlinks. Matemáticamente, el PageRank consiste en calcular el vector propio asociado al valor propio dominante de una matriz de transición estocástica $G \in \mathbb{R}^{n \times n}$, una matriz no negativa en la que cada columna suma 1, representando probabilidades de transición entre nodos. Este vector describe la distribución de probabilidades de un caminante aleatorio, es decir, un modelo que simula a un usuario navegando por la red al moverse entre páginas siguiendo los enlaces disponibles.

Sea $S \in \mathbb{R}^{n \times n}$ una matriz estocástica por columnas. El modelo de transición utilizado para PageRank se define por

$$G = \alpha S + \frac{(1-\alpha)}{n} v_n v_n^T, \quad \alpha \in (0,1), \tag{3.4.1}$$

donde α es un parámetro de amortiguamiento que modela la probabilidad de que un usuario siga un enlace en lugar de saltar a una página aleatoria, $v_n \in \mathbb{R}^n$ es un vector columna de unos, y $\frac{(1-\alpha)}{n} v_n v_n^T$ asegura que la matriz sea estocástica incluso si alguna página no tiene enlaces salientes, [BCRS].

El algoritmo PageRank va mucho más allá de los motores de búsqueda. Específicamente, puede utilizarse para analizar la estructura y las relaciones dentro de redes sociales, como las representadas en el conjunto de datos con el que trabajaremos en este ejemplo numérico. En particular, usaremos los datos de la red social Facebook extraídos del conjunto *ego-Facebook* de

la Stanford Large Network Dataset Collection, [e-F], que representan relaciones de amistad en la red social.

En este caso, cada nodo en el grafo representa un perfil de un usuario, mientras que los enlaces entre nodos indican relaciones de amistad. La matriz de transición estocástica G modela la probabilidad de que un caminante aleatorio, que simula un usuario moviéndose por la red social, pase de un perfil a otro a través de sus conexiones directas. De este modo, el vector propio asociado al valor propio dominante de G proporciona una medida de centralidad o importancia relativa de cada usuario dentro de la red social, similar al modo en que el PageRank mide la relevancia de las páginas web.

Esta capacidad de modelar redes arbitrarias mediante el algoritmo PageRank permite su aplicación en una amplia variedad de contextos, como la identificación de nodos influyentes en redes sociales, la detección de comunidades o la priorización de información en grafos de gran escala. Por lo tanto, aunque la red *ego-Facebook* difiere en naturaleza de una red de páginas web, el algoritmo sigue siendo aplicable y proporciona información valiosa sobre la estructura de la red y la importancia relativa de sus nodos.

Formulación como iteración de punto fijo

El objetivo es encontrar el vector de Perron $u^* \in \mathbb{R}^n$, que es el correspondiente al valor propio $\lambda = 1$, resolviendo

$$Gu = u, \quad \text{con } u \geq 0 \text{ y } \|u\|_1 = 1, \tag{3.4.2}$$

donde $\|\cdot\|_1$ representa la norma ℓ_1. Para resolver este problema numéricamente, lo podemos expresar como una iteración de punto fijo de la forma

$$u_{k+1} = g(u_k),$$

donde $g(u) = Gu$. Comenzamos con un vector inicial u_0 estocástico (es decir, $\|u_0\|_1 = 1$ y con componentes no negativas) y actualizamos iterativamente hasta que $\|g(u_k) - u_k\|_2 < \text{tol}$, con tol una tolerancia fijada.

La convergencia del método es lineal, con un factor de convergencia dado por el cociente de los valores propios dominante y subdominante de G, que se aproxima a $O(\alpha^k)$ a medida que $\alpha \to 1$. Este comportamiento puede ser insuficiente para aplicaciones prácticas, especialmente en redes con millones de nodos, motivando el uso de aceleraciones como AA.

Implementación

Para comparar el método de iteración de punto fijo y la aceleración de Anderson (AA) en el problema de PageRank, hemos desarrollado un programa en MATLAB que analiza la convergencia de ambos métodos. El objetivo es comparar, para distintos valores de m, el comportamiento del residuo $\|g(u_k) - u_k\|_2$ en función del número de iteraciones, tanto para el método de iteración de punto fijo como para AA.

La matriz S se construye directamente a partir del conjunto de datos, que incluye un total de $n = 4039$ nodos. Estos describen las conexiones entre los nodos (amigos en Facebook) en forma de pares ordenados (i, j), donde i y j son los identificadores de los nodos. A partir de estos datos, seguimos los siguientes pasos para construir la matriz estocástica S:

- Primero, se crea una matriz de adyacencia dispersa A, que es una representación matemática de un grafo en forma de matriz. Se utiliza para describir las conexiones (o aristas) entre los nodos (o vértices) de un grafo, donde $A(i, j) = 1$ si hay un enlace de j a i.

- Posteriormente, se manejan los nodos colgantes (columnas sin conexiones salientes) asignando probabilidades uniformes a todas las columnas que suman cero. Esto asegura que la matriz sea válida para el algoritmo de PageRank.

- Finalmente, las columnas de A se normalizan para que cada una sume 1, obteniendo así una matriz S que es estocástica por columnas. Este procedimiento garantiza que S sea apta para el cálculo iterativo en el problema de PageRank.

El programa utiliza como punto de partida un vector inicial u_0, definido como estocástico uniforme: $u_0 = \frac{1}{n}v_n$, donde v_n es un vector columna de unos de tamaño n. Además, se establece una tolerancia de convergencia tol $= 10^{-10}$, que determina el criterio de parada. Cuando el residuo $\|g(u_k) - u_k\|_2$ cae por debajo de esta tolerancia, entonces el bucle se detiene y u_k se considera el vector numérico de Perron del método.

Una vez inicializado el programa, se crea G a partir de (3.4.1), donde hemos tomado $\alpha =$ 0,99.

En el caso de la iteración de punto fijo, el programa implementa un esquema iterativo, donde en cada iteración k se evalúa $u_{k+1} = g(u_k)$ y se calcula el residuo. En la iteración k-ésima se almacena únicamente u_{k+1}, necesario para obtener u_{k+2} en la iteración siguiente, lo que permite un manejo eficaz de la memoria. Si el residuo cae por debajo de la tolerancia antes de alcanzar el número máximo de iteraciones (en nuestro caso, 10), el proceso se detiene.

Para implementar la aceleración de Anderson, utilizaremos la versión que resuelve el problema de mínimos cuadrados sin restricciones (algoritmo 5) para diferentes valores de m. Dado que en cada iteración es necesario calcular $\mathcal{F}_k = \begin{bmatrix}\Delta f_{k-m_k} & \cdots & \Delta f_{k-1}\end{bmatrix}$, se guardan los cálculos con el fin de evitar redundancias y no ralentizar el programa. Esto se debe a que, para determinar $\mathcal{F}_{k+1}$, se pueden reutilizar $m_k - 1$ componentes de $\mathcal{F}_k$ que ya fueron calculadas en la iteración k.

Finalmente, el programa genera una gráfica que compara la norma del residuo k-ésimo frente a k para los distintos métodos. Para este ejemplo usaremos los valores $m = 1, 2, 4$.

Resultados

El gráfico 3.1 muestra la comparación entre el método de iteración de punto fijo y la aceleración de Anderson (AA). Para la primera iteración, todos los métodos aproximan u_1 por $g(u_0)$, pero a partir del segundo paso empiezan a distanciarse, especialmente a partir de la tercera iteración. Para $m = 1$, la norma del residuo es muy parecida. Esto probablemente se debe a que tanto el método de punto fijo como AA con $m = 1$, aproximan u_{k+1} en función de u_k, sin utilizar información de iteraciones previas. Sin embargo, para $m = 2$ y $m = 4$ sí que lo hacen, y la reducción de los residuos es considerablemente más rápida, especialmente en las iteraciones iniciales.

Un aspecto notable es que cuanto mayor es m, el método AA alcanza el mismo nivel de residuo en menos iteraciones, lo que demuestra la ventaja de incluir información de pasos previos. Para el límite impuesto al número de iteraciones, todos los métodos alcanzan la tolerancia fijada para el residuo, lo que ilustra numéricamente su convergencia. Los resultados demuestran que la aceleración de Anderson mejora la velocidad de convergencia, que será mayor cuanto más grande sea m. Además, la mejora se ralentiza según aumenta m, lo que sugiere utilizar valores moderados que aceleren el proceso sin aumentar en exceso el coste computacional.

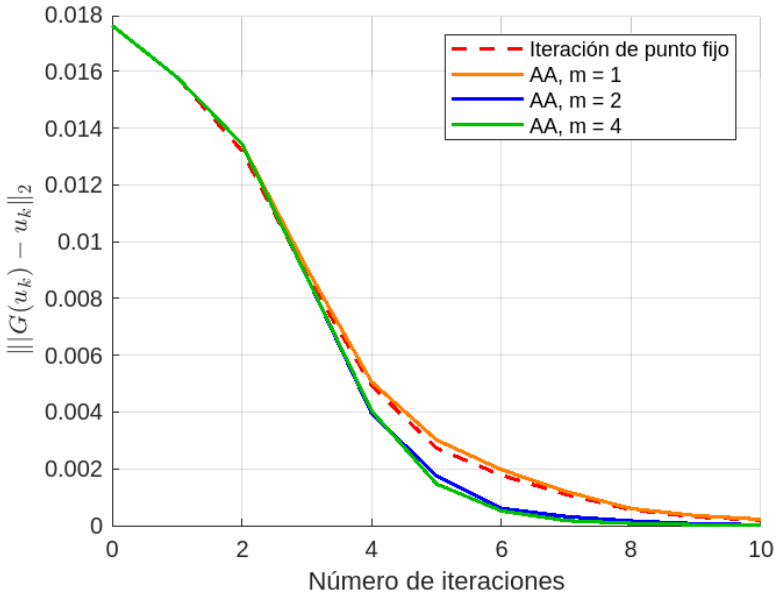


Gráfico 3.1: Comparación entre IPF y AA para $m = 1, 2, 4$.

3.4.2. Segundo ejemplo: Problema de Poisson no lineal

Consideremos, como segundo ejemplo, el siguiente problema de valor inicial y de contorno no lineal

$$\begin{cases} -\left(q(u)\,u'\right)' + r(u) + u' = s(x), \quad x \in (0,1), \\ u(0) = v_0,\ u(1) = v_1, \end{cases} \tag{3.4.3}$$

donde las funciones q, r y s son conocidas, y v_0, v_1 son los valores en los extremos del dominio. Supongamos que (3.4.3) tiene solución única u. Nuestro objetivo es aproximarla discretizando el problema con diferencias finitas y usando para su resolución la aceleración de Anderson.

Discretización y formulación como iteración de punto fijo

Dividimos el intervalo $[0,1]$ en $N+1$ puntos equiespaciados $x_j = jh$, con $h = \frac{1}{N}$. Denotamos $u_j \approx u(x_j)$ como la aproximación a la solución en los nodos x_j, donde $j = 0, \ldots, N$. Las derivadas primera y segunda en x_j se aproximan mediante diferencias finitas de la siguiente manera:

$$u'(x_j) \approx \frac{u_{j+1} - u_{j-1}}{2h}, \quad u''(x_j) \approx \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2}.$$

Sustituyendo estas expresiones en (3.4.3), obtenemos la ecuación discreta en el nodo x_j:

$$-q'(u_j)\left(\frac{u_{j+1} - u_{j-1}}{2h}\right)^2 - q(u_j)\frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + r(u_j) + \frac{u_{j+1} - u_{j-1}}{2h} = s_j,$$

donde $s_j = s(x_j)$ para $j = 1, \ldots, N-1$.

Las condiciones de contorno $u_0 = v_0$ y $u_N = v_1$ se incorporan directamente en la discretización, lo que nos lleva a resolver un sistema no lineal en las incógnitas $u_1, \ldots, u_{N-1}$. El sistema discreto resultante se puede escribir como

$$f(U) = 0,$$

donde $U = \left(u_1, \ldots, u_{N-1}\right)^T$, y $f = \left(f_1, \ldots, f_{N-1}\right)^T : \mathbb{R}^{N-1} \to \mathbb{R}^{N-1}$ está dado por

$$f_j(U) = -\frac{q'(u_j)}{4}(u_{j+1} - u_{j-1})^2 - q(u_j)(u_{j+1} - 2u_j + u_{j-1}) + h^2 r(u_j) + \frac{h}{2}(u_{j+1} - u_{j-1}) - h^2 s_1,$$

para $j = 2, \ldots, N-2$, y extremos dados por

$$\begin{aligned} f_1(U) &= -\frac{q'(u_1)}{4}(u_2 - v_0)^2 - q(u_1)(u_2 - 2u_1 + v_0) + h^2 r(u_1) + \frac{h}{2}(u_2 - v_0) - h^2 s_1, \\ f_{N-1}(U) &= -\frac{q'(u_{N-1})}{4}(v_1 - u_{N-2})^2 - q(u_{N-1})(v_1 - 2u_{N-1} + u_{N-2}) + h^2 r(u_{N-1}) \\ &\quad + \frac{h}{2}(v_1 - u_{N-2}) - h^2 s_1. \end{aligned}$$

Finalmente, considerando la función de iteración $g(U) = f(U)+U$, podemos escribir la iteración de punto fijo como

$$U^{(k+1)} = g(U^{(k)}).$$

Implementación

Para comparar el método clásico de iteración de punto fijo y la aceleración de Anderson, utilizaremos de nuevo un programa en MATLAB. Primero calcularemos la función g a partir de los datos iniciales q, r, s, v_0 y v_1. Para este ejemplo, hemos usado las condiciones iniciales $v_0 = v_1 = 0$ y las funciones

$$\begin{aligned} q(u) &= 1 + u^2, \\ r(u) &= 0, \\ s(x) &= 9\pi^2 \sin(3\pi x) + 9\pi^2 \sin^3(3\pi x) \\ &\quad - 18\pi^2 \cos^2(3\pi x)\sin(3\pi x) + 3\pi \cos(3\pi x). \end{aligned}$$

Estas se han tomado tal que $u(x) = \sin(3\pi x)$ sea la solución exacta. Nuestra aproximación inicial será el vector nulo y ejecutaremos el programa para $m = 3, 5, 6, 10, 20, 40$. A partir de este punto, la implementación es análoga al anterior ejemplo para ambos métodos. Una vez ejecutado el programa y calculados los residuos para cada uno, graficamos los resultados presentando la norma del residuo frente al número de iteraciones máximas permitidas. También graficaremos en $[0, 1]$ la aproximación obtenida $U^{(k)}$ tras 200 iteraciones frente a la solución real del problema $u(x) = \sin(3\pi x)$ para AA con $m = 10, 20, 40$.

Resultados

El gráfico 3.2 muestra la comparación de la norma del residuo $\|f(U^{(k)})\|_2$ frente al número de iteraciones permitidas para los distintos valores de m. Se observa que la iteración de punto fijo diverge rápidamente, como lo demuestra el crecimiento continuo del residuo. Cuando analizamos los resultados obtenidos con la aceleración de Anderson, podemos observar que la divergencia va atenuándose a medida que aumenta m (la norma de los residuos va decreciendo), transformando, a partir de $m = 10$, la divergencia en convergencia. La dificultad de la iteración (el iterante inicial está lejos de la solución y el método clásico es divergente) se refleja en la necesidad de aumentar m, con el consiguiente coste en número de iteraciones.

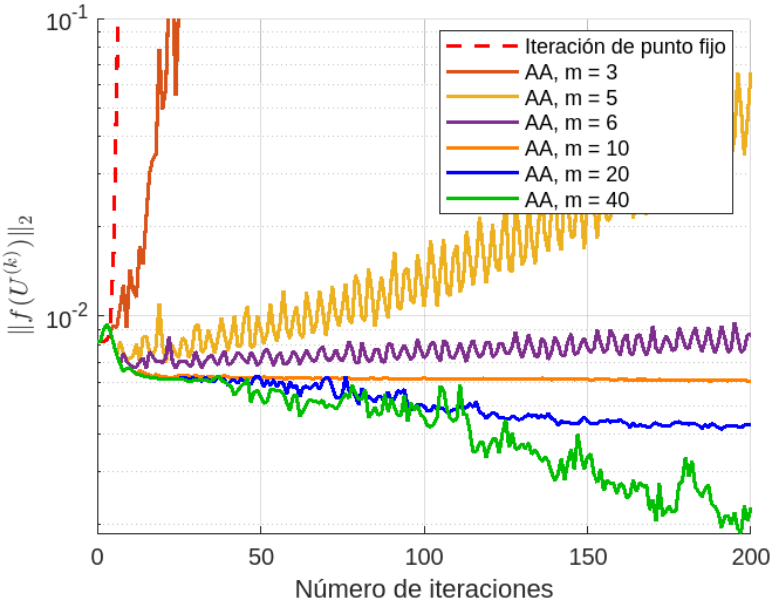


Gráfico 3.2: Comparación de la norma del residuo entre IPF y AA para distintos valores de m.

Asimismo, el gráfico 3.3 muestra la comparación entre la solución exacta $u(x) = \sin(3\pi x)$, dibujada en negro con una línea más gruesa, y las aproximaciones obtenidas tras 200 iteraciones del método AA para $m = 10, 20, 40$, donde se evidencia la mejora en la precisión del método al aumentar m.

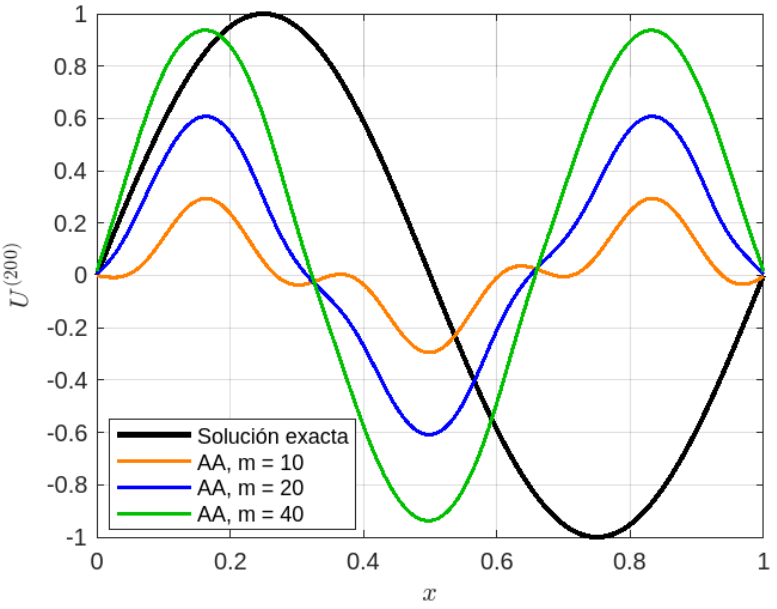


Gráfico 3.3: Comparación entre la solución exacta $u(x) = \sin(3\pi x)$ y las aproximaciones obtenidas con AA para $m = 10, 20, 40$ tras 200 iteraciones.

3.4.3. Tercer ejemplo: Ondas no lineales localizadas

Para el último ejemplo, se considera el problema diferencial

$$-c\phi(z) + \frac{1}{2}\phi(z)^2 + \phi''(z) = 0, \quad z \in \mathbb{R}, \tag{3.4.4}$$

para $c > 0$ fijo, con dos soluciones: $\phi = 0$ y

$$\phi(z) = 3c\,\mathrm{sech}^2\left(\frac{\sqrt{c}}{2}z\right). \tag{3.4.5}$$

Ambas soluciones pueden verse como puntos fijos cuando (3.4.4) se escribe en la forma

$$(c - \partial_x^2)\phi = \frac{1}{2}\phi^2. \tag{3.4.6}$$

Nuestro objetivo es analizar numéricamente (3.4.4) como problema de punto fijo (3.4.6), mediante procesos iterativos. Dada la forma localizada de la solución (3.4.5), el tratamiento numérico requiere primero aproximar (3.4.4) por el problema periódico

$$\begin{cases} -c\phi(z) + \frac{1}{2}\phi(z)^2 + \phi''(z) = 0, \quad z \in [-T, T], \\ \phi(-T) = \phi(T). \end{cases}$$

en un intervalo $[-T, T]$, con T suficientemente grande.

Discretización y formulación como iteración de punto fijo

Dividimos el intervalo $[-T, T]$ en $N + 1$ puntos equiespaciados $z_j = -T + jh$, donde $j = 0, \ldots, N$, y $h = \frac{2T}{N}$. Definimos $\phi_j \approx \phi(z_j)$ como la aproximación a la solución en los nodos z_j. Incorporando las condiciones periódicas en los extremos, la discretización mediante diferencias finitas conduce a las ecuaciones en el nodo z_j

$$\frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{h^2} - c\phi_j + \frac{1}{2}\phi_j^2 = 0,$$

para $j = 0, \ldots, N-1$. Estas condiciones se traducen en las ecuaciones discretas

$$\begin{aligned} &\phi_1 - (2 + ch^2)\phi_0 + \phi_{N-1} = -\frac{h^2}{2}\phi_0^2, && j = 0, \\ &\phi_{j+1} - (2 + ch^2)\phi_j + \phi_{j-1} = -\frac{h^2}{2}\phi_j^2, && j = 1, \ldots, N-2, \\ &\phi_0 - (2 + ch^2)\phi_{N-1} + \phi_{N-2} = -\frac{h^2}{2}\phi_{N-1}^2, && j = N-1. \end{aligned} \tag{3.4.7}$$

Reformulamos como un sistema matricial. Sea el vector $\Phi = \left(\phi_0, \phi_1, \ldots, \phi_{N-1}\right)^T$, que representa las incógnitas en los nodos. Entonces, (3.4.7) puede escribirse de la forma

$$L\Phi = -\frac{h^2}{2}\mathbf{F}(\Phi), \tag{3.4.8}$$

donde L y $\mathbf{F}(\Phi)$ son

$$L = \begin{pmatrix} -2 - ch^2 & 1 & 0 & \cdots & 0 & 1 \\ 1 & -2 - ch^2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & -2 - ch^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & 1 & -2 - ch^2 \end{pmatrix}, \quad \mathbf{F}(\Phi) = \begin{pmatrix} \phi_0^2 \\ \phi_1^2 \\ \vdots \\ \phi_{N-1}^2 \end{pmatrix}.$$

Una vez obtenido el sistema (3.4.8), podemos formular una iteración de punto fijo. Nuestra función de iteración es

$$g(\Phi^{(k)}) = -\frac{h^2}{2}L^{-1}\mathbf{F}(\Phi^{(k)}) = \Phi^{(k+1)}, \tag{3.4.9}$$

aunque en la práctica no necesitamos calcular explícitamente L^{-1}. En lugar de eso, resolveremos el sistema lineal en cada iteración.

Como alternativa al método clásico (3.4.9), se considera la iteración

$$m_k = \frac{\langle L\Phi^{(k)}, \Phi^{(k)}\rangle}{\langle F\left(\Phi^{(k)}\right), \Phi^{(k)}\rangle}, \tag{3.4.10}$$

$$\Phi^{(k+1)} = m_k^2 g(\Phi^{(k)}), \quad k = 0, 1, \ldots, \tag{3.4.11}$$

que es una modificación de (3.4.9) con un factor de estabilización (3.4.10). El método (3.4.11), que llamaremos IPF estabilizado, suele utilizarse en problemas donde la iteración clásica (3.4.9) no es convergente o bien converge pero con inestabilidad o muy lentamente, [PS].

Implementación

Para comparar el método de iteración de punto fijo y la aceleración de Anderson, utilizaremos el programa de MATLAB implementado en el ejemplo anterior, con una única modificación: la definición de la función de iteración. En este caso, emplearemos una malla de $N = 5000$ puntos, con la cual construiremos la matriz L previamente definida. Esta matriz nos permitirá definir la función g, que no implementaremos como se describe en (3.4.9) por motivos de eficiencia computacional. En su lugar, definiremos la función de iteración g tal que, dado $\Phi^{(k)}$, devuelva $\Phi^{(k+1)}$ solución de

$$L\Phi^{(k+1)} = -\frac{h^2}{2}\mathbf{F}(\Phi^{(k)}).$$

Es decir, resolveremos el sistema en cada paso. Este enfoque es mucho menos costoso que calcular explícitamente L^{-1}. En cuanto a la norma del residuo k-esimo, en este problema es

$$\|r_k\|_2 = \|L\Phi^{(k)} + \frac{h^2}{2}\mathbf{F}(\Phi^{(k)})\|_2. \tag{3.4.12}$$

Para este ejemplo, usaremos $\Phi^{(0)} = 3c\left(\operatorname{sech}(z_0), \ldots, \operatorname{sech}(z_{N-1})\right)^T$ como aproximación inicial. Ejecutaremos el programa para los valores $m = 1, 2, 4$ y $c = 3$, $T = 16$.

Resultados

Comenzamos comparando la iteración clásica (3.4.9) con su modificación a partir de la aceleración de Anderson sin estabilización. El gráfico 3.4 compara la norma euclídea del residuo (3.4.12) frente al número de iteraciones para (3.4.9) y AA con distintos valores de m. Como en el ejemplo anterior, tenemos que la iteración clásica es divergente, mientras que, si se implementa con AA, pasa a converger, esta vez más rápidamente (observemos que el iterante inicial está relativamente cerca de (3.4.5)).

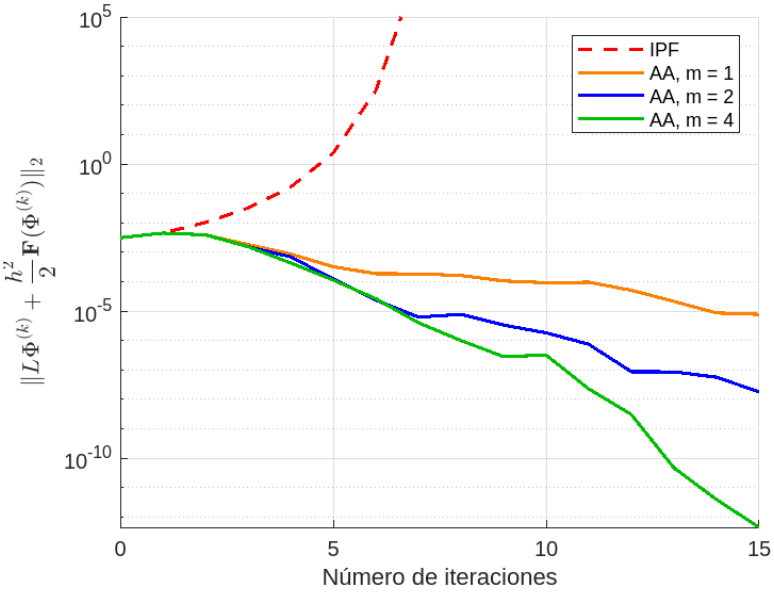
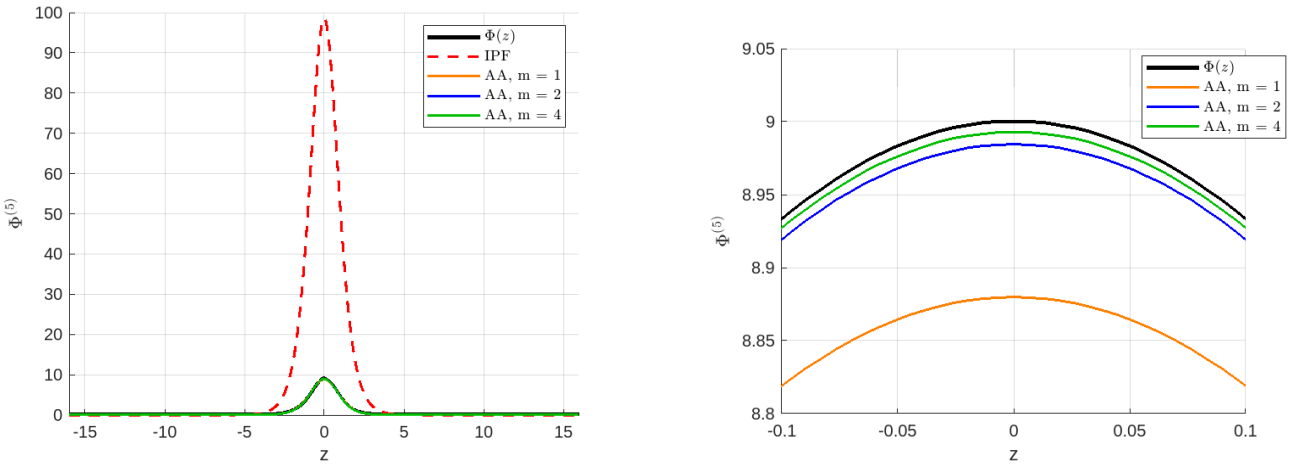


Gráfico 3.4: Comparación de la norma del residuo entre los métodos IPF y AA sin estabilización para distintos valores de m.

El gráfico 3.5a compara las aproximaciones obtenidas por cada método tras 5 iteraciones con (3.4.5), representada en negro. Dado que las diferencias entre AA y la solución no trivial son prácticamente imperceptibles, también mostramos el gráfico 3.5b, que amplía la región alrededor del máximo para apreciar mejor las variaciones. Además se observa que, a partir de un valor, aumentar m no mejora significativamente la aceleración.



(a) Vista general.

(b) Aumento.

Gráfico 3.5: Comparación entre las aproximaciones obtenidas con IPF y AA para $m = 1, 2, 4$ sin estabilización tras 5 iteraciones.

Ahora comparamos con la versión estabilizada. En el gráfico 3.6, observamos que tanto (3.4.11) como AA estabilizado son convergentes. En este caso, AA actúa como acelerador de la convergencia a partir de un valor de m.

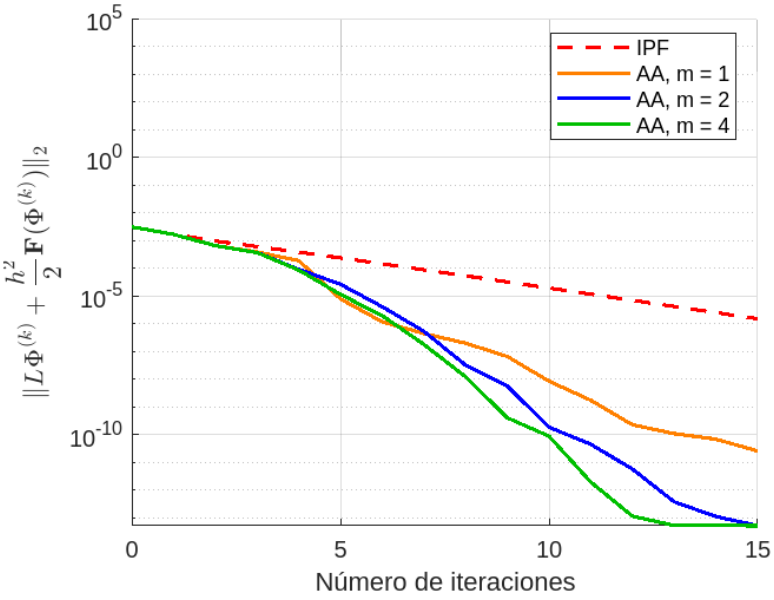
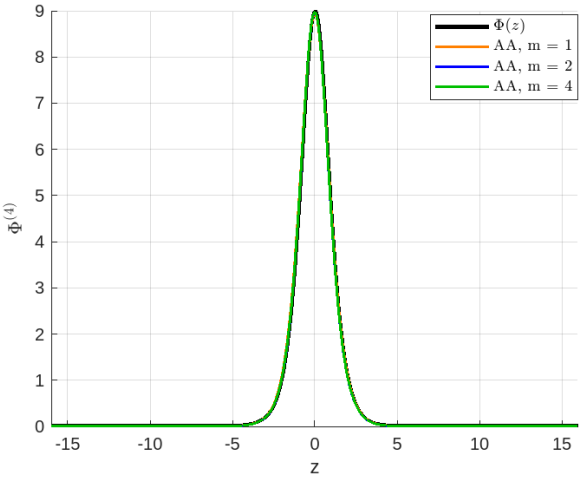
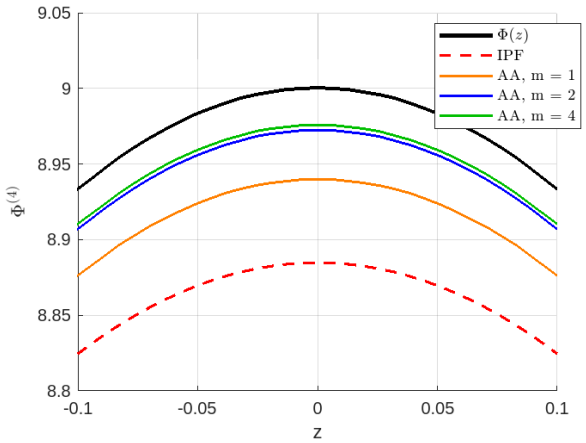


Gráfico 3.6: Comparación de la norma del residuo entre los métodos IPF y AA estabilizados para distintos valores de m.

El gráfico 3.7a compara las aproximaciones obtenidas por cada método estabilizado tras 4 iteraciones con la solución (3.4.5). Dado que las diferencias son prácticamente imperceptibles, también mostramos el gráfico 3.7b, que de nuevo amplía la región alrededor del máximo para apreciar mejor las variaciones. Se puede observar que tanto la solución obtenida con IPF estabilizado como con distintos AA estabilizados (para $m = 1, 2, 4$) convergen hacia una solución no trivial en muy pocos pasos, simulando la forma localizada de la solución (3.4.5). De nuevo se observa que no siempre es beneficioso aumentar m indefinidamente, pues la mejora es mínima.



(a) Vista general.



(b) Aumento.

Gráfico 3.7: Comparación entre las aproximaciones obtenidas con IPF y AA para $m = 1, 2, 4$ estabilizados tras 4 iteraciones.

Una última observación se refiere al carácter local de la convergencia. Con valores de c pequeños, el dato inicial tiene una amplitud lo suficientemente baja como para estar más cercano a la solución $\phi = 0$. En este caso, los experimentos mostraban que las iteraciones (incluida la generada por (3.4.9)) convergían hacia la solución trivial.

Bibliografía

[A] Anderson, D. G. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 1965, vol. 12, no. 4, p. 547-560.

[AR] Arnoldi, W. E. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 1951, vol. 9, no. 1, p. 17-29.

[B] Broyden, C. G. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 1965, vol. 19, p. 577–593.

[BR] Brezinski, C.; Redivo-Zaglia, M. (Eds.). *Extrapolation Methods: Theory and Practice. Université des Sciences et Technologies de Lille, Villeneuve d'Ascq, France; Università degli Studi di Padova, Padova, Italy*, 1991, vol. 2, p. 1-464.

[BCRS] Brezinski, C.; Cipolla, S.; Redivo-Zaglia, M.; Saad, Y. Shanks and Anderson-type acceleration techniques for systems of nonlinear equations. *IMA Journal of Numerical Analysis*, 2022, vol. 42, no. 4, p. 3058–3093.

[BC] Bierlaire, M.; Crittin, F. Solving noisy, large-scale fixed-point problems and systems of nonlinear equations. *Transportation Science*, 2006, vol. 40, no. 1, p. 44-63.

[D] Demmel, J. *Numerical Linear Algebra*. Center for Pure and Applied Mathematics, Department of Mathematics, University of California, 1993.

[DM] Dennis, J. E.; Moré, J. J. Quasi-Newton methods: motivation and theory. *SIAM Review*, 1977, vol. 19, no. 1, p. 46-89.

[e-F] McAuley, J.; Leskovec, J. Learning to discover social circles in ego networks. *Stanford Large Network Dataset Collection*, 2012. Disponible en: `https://snap.stanford.edu/data/ego-Facebook.html`.

[EN] Eirola, T.; Nevanlinna, O. Accelerating with rank-one updates. *Linear Algebra and its Applications*, 1989, vol. 121, p. 511–520.

[F] Ford, W. *Numerical Linear Algebra with Applications: Using MATLAB*. Academic Press, 2014.

[FS] Fang, H.; Saad, Y. Two classes of multisecant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications*, 2009, vol. 16, p. 197–221.

[JS] Jbilou, K.; Sadok, H. Vector extrapolation methods: applications and numerical comparison. *Journal of Computational and Applied Mathematics*, 2000, vol. 122, no. 1-2, p. 149-165.

[K] Khatiwala, S. Efficient spin-up of Earth system models using sequence acceleration. *Science Advances*, 2024, vol. 10, no. 18, p. eadn2839.

[L] Lorentzon, G. A relation between Anderson acceleration and GMRES. *Bachelor's Theses in Mathematical Sciences*, 2020.

[MM] Morton, K. W.; Mayers, D. F. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, 1995.

[S] Saad, Y. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2003.

[SI] Sidi, A. *Vector Extrapolation Methods with Applications*. Society for Industrial and Applied Mathematics, 2017.

[SS] Sanz-Serna, J. M. *Diez lecciones de cálculo numérico*. Universidad de Valladolid, Secretariado de Publicaciones e Intercambio Editorial, 2010.

[PS] Pelinovsky, D.; Stepanyants, Y. Convergence of Petviashvili's iteration method for numerical approximation of stationary solutions of nonlinear wave equations. *SIAM Journal on Numerical Analysis*, 2004, vol. 42, no. 1, p. 1110-1127.

[TK] Toth, A.; Kelley, C. T. Convergence analysis for Anderson acceleration. *SIAM Journal on Numerical Analysis*, 2015, vol. 53, no. 2, p. 805-819.

[W] Woodbury, M. A. Inverting modified matrices. *Report Memorandum 42, Statistical Research Group, Princeton University*, 1950.

[WN] Walker, H. F.; Ni, P. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 2011, vol. 49, no. 4, p. 1715-1735.