

Métodos Numéricos

# Métodos Iterativos para el Problema de Valores Propios

## Tarea 6

September 30, 2020

Erika Rivadeneira Pérez  
*erika.rivadeneira@cimat.mx*  
*Matemáticas Aplicadas - CIMAT*

## 1 Resumen

En el presente reporte se exponen métodos iterativos que producen, en un paso  $k$ , un autovector  $v_k$  aproximado asociado con un autovalor aproximado  $\lambda_k$  que converge al autovector  $v$  deseado y al autovalor  $\lambda$  a medida que aumenta el número de iteraciones. Se consideran tres métodos iterativos llamados método de potencia inversa, método de potencia inversa con deflación y método de Jacobi.

## 2 Introducción

La determinación de autovalores y autovectores de una matriz cuadrada  $A$  de orden  $n$  es un problema que se presenta en numerosas ramas de la Matemática. Dado que los autovalores son las raíces del polinomio característico

$$P(\lambda) = \det(A - \lambda I) = (-1)^n [\lambda^n + a_1 \lambda^{n-1} + \dots + a_{n-1} \lambda + a_n]$$

cualquier método para calcular valores propios implica necesariamente un número infinito de pasos. Esto se debe a que encontrar valores propios de una matriz  $n \times n$  es equivalente a encontrar las raíces de su polinomio característico de grado  $n$  y calcular los coeficientes del polinomio característico requiere el cálculo del determinante, sin embargo, el problema de encontrar las raíces de un polinomio puede ser muy mal condicionado y para  $n > 4$  tales raíces no se pueden encontrar, por tanto, es claro que los métodos de cálculo de autovalores deben ser iterativos [1]. En este caso particular, para el cálculo de autovalores  $\lambda$  se utilizan tres métodos iterativos, método de la potencia inversa normal y con deflación y el método de Jacobi

## 3 Metodología

### 3.1 Método de la Potencia Inversa

El método de la potencia Inversa es una modificación del método de la potencia que ofrece una convergencia más rápida. Se usa para determinar el valor propio de  $A$  más cercano a un número  $q$  específico. Supongamos que la matriz  $A$  tiene valores propios  $\lambda_1, \dots, \lambda_n$  con vectores propios linealmente independientes  $v^{(1)}, \dots, v^{(n)}$ . Consideremos la matriz  $(A - qI)^{-1}$ , donde  $q \neq \lambda_i$  para  $i = 1, 2, \dots, n$ . Los valores propios de  $(A - qI)^{-1}$  son

$$\frac{1}{\lambda_1 - q}, \quad \frac{1}{\lambda_2 - q}, \quad \dots, \quad \frac{1}{\lambda_n - q}$$

con vectores propios  $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(n)}$ . Al aplicar el método de la potencia en  $(A - qI)^{-1}$  obtenemos

$$y^{(m)} = (A - qI)^{-1} x^{(m-1)}$$

$$\mu^{(m)} = y_{p_{m-1}}^{(m)} = \frac{y_{p_{m-1}}^{(m)}}{y_{p_{m-1}}^{(m-1)}} \quad (1)$$

$$= \frac{\sum_{j=1}^n \beta_j \frac{1}{(\lambda_j - 1)^m v_{p_{m-1}}^{(j)}}}{\sum_{j=1}^n \beta_j \frac{1}{(\lambda_j - 1)^{m-1} v_{p_{m-1}}^{(j)}}} \quad (2)$$

y

$$x^{(m)} = \frac{y^{(m)}}{y_{p_m}^{(m)}}$$

donde  $p_m$  representa en cada paso el entero más pequeño para el cual  $|y_{p_m}^{(m)}| = \|y^{(m)}\|_\infty$ . La sucesión  $\{\mu^{(m)}\}$  de (1) converge a  $\frac{1}{(\lambda_k - q)}$ , donde

$$\frac{1}{|\lambda_k - q|} = \max_{1 \leq i \leq n} \frac{1}{|\lambda_i - q|}$$

y  $\lambda_k = q + \frac{1}{\mu^{(m)}}$  es el valor propio de  $A$  mas cercano a  $q$ . Cuando  $k$  se conoce, la ecuacion (1) puede escribirse como

$$\mu^{(m)} = \frac{1}{\lambda_k - q} \left[ \frac{\beta_k v_{p_{m-1}}^{(k)} + \sum_{j=1, j \neq k}^n \beta_j \left[ \frac{\lambda_k - q}{\lambda_j - 1} \right]^m v_{p_{m-1}}^{(j)}}{\beta_k v_{p_{m-1}}^{(k)} + \sum_{j=1, j \neq k}^n \beta_j \left[ \frac{\lambda_k - q}{\lambda_j - 1} \right]^{m-1} v_{p_{m-1}}^{(j)}} \right]$$

Por tanto, la eleccion de  $q$  determina la convergencia siempre y cuando  $\frac{1}{(\lambda_k - q)}$  sea un valor propio dominante y único de  $(A - qI)^{-1}$ . Cuanto más se acerque  $q$  a un valor propio  $\lambda_k$  de  $A$ , mas rápida será la convergencia.

Por otro lado,  $q$  se calcula a partir de una aproximación inicial  $\mathbf{x}^{(0)}$  de un vector propio por medio de

$$q = \frac{\mathbf{x}^{(0)'} A \mathbf{x}^{(0)}}{\mathbf{x}^{(0)'} \mathbf{x}^{(0)}}$$

Esta eleccion de  $q$  proviene de la observacion de que, si  $x$  es un vector propio de  $A$  respecto al valor propio  $\lambda$ , entonces  $Ax = \lambda x$ . Por tanto,  $x'Ax = \lambda x'x$  y

$$\lambda = \frac{x'Ax}{x'x} = \frac{x'Ax}{\|x\|_2^2}$$

Por otro lado, se encuentran disponibles numerosas técnicas para obtener aproximaciones a los otros valores propios de una matriz una vez que se ha calculado una aproximación al valor propio dominante. Una de estas técnicas es conocida como la técnica de deflación. Las *técnicas de deflación* implican formar una nueva matriz  $B$  cuyos valores propios son los mismos que los de  $A$ , excepto que el valor propio dominante de  $A$  se reemplaza por el valor propio 0 en  $B$  [2].

### 3.2 Método de Jacobi

El algoritmo de valores propios de Jacobi es un método iterativo para el cálculo de los valores propios y los vectores propios de una matriz simétrica.

Primero consideremos la siguiente rotación llamada *rotación de Givens*, su forma matricial es

$$G(i, j, \theta) = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & c & \cdots & -s & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & s & \cdots & c & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \quad (3)$$

donde  $c = \cos \theta$ ,  $s = \sin \theta$  intersección de  $i$ -ésima y  $j$ -ésima filas y columnas. Es decir, los elementos distintos de cero de la matriz de Givens están dados por  $g_{kk} = 1$  para  $k \neq i, j$

$$\begin{aligned} g_{ii} &= c \\ g_{jj} &= -s \\ g_{ij} &= s \text{ para } i > j \end{aligned}$$

El método de Jacobi se basa en que existen matrices ortogonales  $P$ , tales que transforman a la matriz  $A$  en una matriz cuya diagonal principal está formada por los valores propios.

$$\mathbf{PAP}^t = \begin{pmatrix} \lambda_0 & 0 & \cdots 0 \\ 0 & \lambda_1 & \cdots 0 \\ \cdots & \cdots & \cdots \\ 0 & 0 & \cdots \lambda_{n-1} \end{pmatrix}$$

Estas matrices  $P$  tienen la forma de la matriz de rotación (3) donde todos los elementos son nulos, excepto la diagonal principal y los elementos  $(i, j)$  y su simétrico  $(j, i)$ . Dado que  $P$  es ortogonal la matriz  $B$  es tal que  $B = PAP^t$  tiene los mismos valores propios que  $A$ , y además eligiendo convenientemente el ángulo  $\theta$  se puede conseguir que los elementos  $b_{ij}$  y  $b_{ji}$  sean nulos, los valores de dichos ángulos son:

$$\begin{aligned} \cos \theta &= \sqrt{\frac{z+y}{2z}} \\ \sin \theta &= v \sqrt{\frac{z-y}{2z}} \\ v &= \frac{x/y}{|x/y|} \end{aligned}$$

con

$$\begin{aligned} x &= 2a_{kl} \\ y &= a_{ii} - a_{jj} \\ z &= \sqrt{x^2 + y^2} \end{aligned}$$

de esta manera se obtiene una matriz  $B$  más sencilla que  $A$  con los mismos valores propios.

En el método de Jacobi, se ha de encontrar una sucesión de matrices ortogonales  $P$  tal que hagan la matriz  $A$  diagonal. Así,

$$\mathbf{B} = \mathbf{P}_m \mathbf{P}_{m-1} \cdots \mathbf{P}_2 \mathbf{P}_1 \mathbf{A} \mathbf{P}_1^t \mathbf{P}_2^t \cdots \mathbf{P}_{m-1}^t \mathbf{P}_m^t$$

El proceso se concluye cuando el máximo valor absoluto de los elementos fuera de la diagonal de  $B$  pueden ser considerados despreciables frente al valor absoluto de los elementos de la diagonal principal.

Las columnas de la matriz

$$\mathbf{Q} = \mathbf{P}_1^t \mathbf{P}_2^t \cdots \mathbf{P}_{m-1}^t \mathbf{P}_m^t$$

son los vectores propios asociados a los valores propios  $b_{00}, b_{11}, \dots, b_{n-1n-1}$  que constituyen los elementos de la matriz diagonal  $B$ .

## 4 Implementación

A continuación se presentan los pseudo-códigos del método de la potencia inversa (1), del método de la potencia inversa con deflación (2) y el método de Jacobi (3). Para el algoritmo (1) y 3 se requiere de una matriz  $A_{n \times n}$ , un número máximo de iteraciones y una tolerancia o precisión como datos de entrada. Por otro lado, el algoritmo (2) requiere un argumento extra de entrada, el número de autovalores más pequeños que se desea encontrar, el cual está denotado por  $r$ .

---

**Algorithm 1:** Algoritmo de Método de la Potencia Inversa

---

```
Result: lambda, v1 iteracion/* Autovalor más pequeño, autovector asociado y número
de iteraciones */
n,m=shape(A);/* extraemos la dimensión de la matriz */
vo = random.rand(n);
q = (vo'*A*vo)/(vo'*vo);
cont = 1;
vop = argmin(vo)/* argumento más grande de vo */
vo=vo/vop;
while cont < max - iter do
    Solve((A-q*I)v1=x);
    lambda = v1p /* argumento más grande de v1 */
    error =||vo - (v1/v1p)||;
    vo=v1/v1p;
    if error< tol then
        | lambda=(1/lambda)+q
    end
    cont+=cont
end
```

---

---

**Algorithm 2:** Algoritmo de Método de la Potencia Inversa con Deflación

---

```
Result: w, v/* Vector de autovalores más pequeños y matriz de autovectores
asociados */
n,m=shape(A);/* extraemos la dimensión de la matriz */
num=0;
w = []/* Inicializamos matriz para guardar autovalores */
v = []/* Inicializamos matriz para guardar autovectores */
error = 1 /* Inicializamos error */
cont = 0 /* Inicializamos contador */
vo = random.rand(n);
q = (vo'*A*vo)/(vo'*vo);
for i in (1:r) do
    suma=0;
    while cont < max - iter do
        Solve((A-q*I)v1=x);
        lambda = v1p /* argumento más grande de v1 */
        error =||vo - (v1/v1p)||;
        vo=v1/v1p;
        if error< tol then
            | lambda=(1/lambda)+q
        end
        cont+=cont;
        w = w.append(lambda);
        v = v.append(v1)
    end
end
```

---

Tanto el algoritmo (1) y (2) incluyen un vector inicial el cual es un vector aleatorio.

---

**Algorithm 3:** Algoritmo de Método de Jacobi para autovalores

---

```
Result: Autovalores, Autovectores
n,m=shape(A);/* extraemos la dimensión de la matriz */
num=0;
D=A;
P=I;
d[ij] = argmax(D[ij]);/* extraemos el argumento máximo fuera de la diagonal */
if d[ii]=d[jj]/* Encontramos el ángulo theta */
  then
    if d[ij] > 0 then
      | theta=pi/4
    end
    else
      | theta=-pi/4
    end
  end
else
  | theta=1/2*tan-1(2d[ij]/(d[ii] - d[jj]))
end
p[ij]=0;
p[kk]=1/* Ponemos 1's la diagonal */
;
/* Empiezan las rotaciones */
p[ii] =p [jj] =cos;
p[ij] = sin;
p[ji] = sin;
D=P1TDP1;
P=PP1
```

---

## 5 Resultados

Para el uso de los métodos mencionados se consideraron las matrices

$$A_{3 \times 3} = \begin{bmatrix} 3 & -0.1 & -0.2 \\ -0.1 & 7 & -0.3 \\ -0.2 & -0.3 & 10 \end{bmatrix} \quad \text{y} \quad A_{50 \times 50} = \begin{bmatrix} 10 & \dots & 0.00697553 \\ \vdots & \ddots & \vdots \\ 0.00697553 & \dots & 500 \end{bmatrix}$$

Utilizando el método de la potencia inversa y el método de la potencia inversa con deflación se obtuvieron los autovalores más pequeños en valor absoluto y sus respectivos autovectores mostrados a continuación en la tabla (1). Estos resultados están divididos en dos partes. La primera parte muestra solamente el autovector más pequeño con su autovector asociado. En la segunda parte se aprecian los 2 autovectores más pequeños de la matriz  $A_{3 \times 3}$  y el primer y décimo autovalor de la matriz  $A_{50 \times 50}$ . En todos estos casos con su respectivo autovector asociado.

El vector inicial escogido en ambos métodos fue un vector aleatorio. El número máximo de iteraciones considerado en todos los casos es de 10000 iteraciones. **Iter** representa el número de iteraciones realizadas para encontrar el autovalor y **Tol** es la tolerancia o precisión impuesta, la cual es la misma para todos los casos.

Método	Matriz	Autovalor	Autovector	Error	Tol.	Iter.
<i>Potencia Inversa</i>	$A_{3 \times 3}$	1º 2.991343363407016	[0.99925148, 0.0250918, 0.02944269],	5.459e-08	1e-7	17
	$A_{50 \times 50}$	1º 9.998050193188677	[ 9.99955304e-01... -1.14810660e-05],	4.268e-08	1e-7	19

Table 1 continued from previous page

Método	Matriz	Autovalor	Autovector	Error	Tol.	Iter.
<i>Potencia Inversa con deflación</i>	$A_{3 \times 3}$	1° 2.9913433631613806	[0.99925148, 0.0250918, 0.02944269]'	8.827e-08	1e-7	40
		2° 6.973860483506629	[-0.06484618, 0.99330514, 0.09560268]'	8.827e-08	1e-7	40
	$A_{50 \times 50}$	1° 9.998050189539335	[ 9.99955251e-01,..., -1.14805028e-05]'	8.638e-08	1e-7	141
		10° 99.99943676271664	[3.03235130e-04,..., -7.78833863e-05]'	8.638e-08	1e-7	141
<i>Jacobi</i>	$A_{3 \times 3}$	1° 2.99134332	[ 0.9991909, 0.02714663, 0.02967504]	1.368e-12	1e-7	6
		2° 6.97386041	[-0.02989967, 0.99486909, 0.09665145]	1.368e-12	1e-7	6
		3° 10.03479626	[-0.02689902, -0.09746053, 0.99487582]	1.368e-12	1e-7	6
	$A_{50 \times 50}$	1° 9.99805017	[9.99999548e-01,..., 1.76662508e-05]'	1.317e-08	1e-7	2292
		10° 99.99943662	[-0.000297,..., 0.000082]'	1.317e-08	1e-7	2292
		50° 500.00154272	[-0.000011,..., 0.999961]'	1.317e-08	1e-7	2292

Table 1: Resultados del Método de la Potencia Inversa, Método de la Potencia Inversa con Deflación y Método de Jacobi

Es importante mencionar que para encontrar el autovalor más pequeño en el método de la potencia inversa se utiliza como solver de sistema de ecuaciones al método de Crout debido que este método no tiene tantas limitaciones y admite casi cualquier matriz que sea factorizable.

## 6 Discusión

Aunque el método de Jacobi solo es aplicable para matrices simétricas esta técnica fue muy útil y más exacta al calcular los valores propios con sus respectivos vectores asociados debido a que el error en el caso de la matriz  $A_{3 \times 3}$  es menor con  $1.368e - 12$  al error que se tiene al calcularlos con el método de la potencia con Deflación. Esto puede deberse a que el método de la potencia utiliza funciones adicionales para resolver un sistema de ecuaciones y esto puede volverse costoso cuando la matriz es de una dimensión  $n \times n$  mayor.

Resulta evidente que en el caso de que se requiera del valor propio dominante es preferible usar el método de la potencia inversa. En cualquier caso, al usar cada método, en las matrices que fueron consideradas, se obtuvo convergencia casi inmediata. Esto se puede evidenciar con el número de iteraciones resultantes, el método que más se demoró en converger fue el de Jacobi en la matriz  $A_{50 \times 50}$  con 2292, aún así no se acercó siquiera al límite de iteraciones propuestas de 10000.

## 7 Conclusiones

Utilizar estos métodos iterativos ya sea para encontrar un valor propio dominante, varios o todos los valores propios de una matriz resulta bastante conveniente ya que en cada caso se obtuvo lo esperado con pequeños errores de aproximadamente  $1e - 8$ .

Un inconveniente es que el método de Jacobi se limita a encontrar autovalores y autovectores de matrices simétricas mientras que los métodos de la potencia no tienen estas limitantes, por esta razón se utilizó la descomposición  $LU$  para resolver sistemas de ecuaciones con el objetivo de encontrar los valores deseados aunque para matrices de gran dimensión resulte costoso computacionalmente.

## References

- [1] Álvarez, L. and Martínez, A., 2005. Lecciones De Métodos Numéricos. 5th ed. Universidad de Vigo-DMA, pp.115-120.
- [2] R.L. Burden and J.D. Faires, Análisis Numérico, Grupo Editorial Iberoamérica, México 1985.