

## EL MÈTODE APT (*AUTO ADJUSTING PERTURBATION THEORY*) DE DIAGONALITZACIÓ DE MÀTRIU

E. Besalú, R. Carbó-Dorca

Institut de Química Computacional, Universitat de Girona, plaça Hospital, 6, 17071 Girona. Catalunya. Espanya. A/e: emili@canigo.qo.ub.es

---

### RESUM

Es presenta un nou algorisme per a la diagonalització de matrius amb diagonal dominant. Es mostra la seva eficàcia en el tractament de matrius no simètriques, amb elements definits sobre el cos complex i, fins i tot, de grans dimensions. Es posa de manifest la senzillesa del mètode així com la facilitat d'implementació en forma de codi de programació. Es comenten els seus avantatges i característiques limitants, així com algunes de les millores que es poden implementar. Finalment, es mostren alguns exemples numèrics.

### RESUMEN

Se presenta un nuevo algoritmo para la diagonalización de matrices con diagonal dominante. Se muestra su eficacia en el tratamiento de matrices no simétricas, con elementos definidos en el campo complejo e, inclusive, de grandes dimensiones. Se pone de manifiesto la sencillez del método así como la facilidad de implementación en forma de código de programación. Se comentan sus ventajas y características limitantes, así como alguna de las mejoras que el método puede asumir. Finalmente, se muestran algunos ejemplos numéricos.

### ABSTRACT

A new algorithm is presented which is useful for the diagonalization of diagonal dominant matrices. Its efficacy in the treatment of non-symmetric matrices, which complex elements or, even, of large dimensions is shown. The simplicity of the method and the easiness of programming implementation are also revealed. The method characteristics are also commented together with possible enhancements that can be considered. Finally, some numerical examples are given.

**Keywords:** Diagonalization. Complex Matrices. Perturbation Theory. APT Method. Large Matrices.

---

## FONAMENTS TEÒRICS DEL MÈTODE

El mètode APT (*Autoadjusting Perturbation Theory*) (1) es va presentar inicialment com una nova teoria de perturbacions amb millors característiques de convergència que les ofertes per la teoria de perturbacions de Rayleigh-Schrödinger (1). Aquí es presenta una versió simplificada del mètode, la qual s'aplica a trobar un sol vector i valor propis d'una matriu quadrada *arbitrària* però amb diagonal suficientment dominant.

Molts mètodes de diagonalització de matrius es basen a aplicar transformacions de semblança sobre la matriu que s'ha de tractar. Per exemple, la diagonalització d'una matriu  $\mathbf{H}$  es pot iniciar aplicant la transformació de semblança originada per una matriu  $\mathbf{Z}$ :

$$\mathbf{H}_2 = \mathbf{Z}^{-1} \mathbf{H} \mathbf{Z} \quad (1)$$

i el procés es pot anar iterant sobre la nova matriu obtinguda fins a assolir la completa diagonalització de la matriu de partida (2,3). Si se segueix aquesta metodologia, és ben sabut que per conèixer els vectors propis de la matriu  $\mathbf{H}$  caldrà anar avaluant els successius productes entre les diferents matrius de transformació  $\mathbf{Z}$  que es vagin utilitzant.

Aquest treball se centra en el cas de trobar un sol vector i valor propis de  $\mathbf{H}$ . Per localitzar un valor propi es pot recórrer a anul·lar el radi del disc de Gershgorin (4,5) associat a un element diagonal de la matriu  $\mathbf{H}$ . Això equival a cercar una matriu de transformació  $\mathbf{Z}$  que anul·li tots els elements d'una columna (o filera) excepte el diagonal (2). Aquesta operació l'anomenarem diagonalitzar una columna (o filera) de la matriu  $\mathbf{H}$ . Una vegada s'aconsegueix la diagonalització de la columna, el valor diagonal remanent és el valor propi cercat i la matriu de transformació  $\mathbf{Z}$  conté la informació sobre el vector propi a què està associat.

El mètode APT es basa en la utilització d'una forma particular de les matrius de transformació  $\mathbf{Z}$  de tal manera que només es diagonalitza una sola i predeterminada columna de la matriu  $\mathbf{H}$ . D'altra banda,  $\mathbf{Z}$  té una estructura particular que permet el càlcul senzill i immediat de la seva inversa i, per tant, del doble producte matricial que defineix la transformació de semblança (1). Concretament, quan es vol diagonalitzar la columna  $p$ -èsima de la matriu  $\mathbf{H}$  d'ordre  $n$ , la matriu quadrada  $\mathbf{Z}$  també és d'ordre  $n$  i es defineix com a

$$\mathbf{Z} = \mathbf{I} + \mathbf{W} \quad (2)$$

a on  $\mathbf{I}$  és la matriu identitat i  $\mathbf{W}$  és la matriu nul·la a la qual s'ha substituït la  $p$ -èsima columna per un vector  $\mathbf{w}$  que té la  $p$ -èsima component també nul·la.

A efectes pràctics només cal considerar la forma explícita del vector  $\mathbf{w}$  o  $p$ -èsima columna de la matriu  $\mathbf{W}$ :

$$\mathbf{w} = \{ w_i \mid w_p = (0,0) \} \quad (3)$$

a on  $(0,0)$  és l'element neutre per a la suma en el cos complex.

D'altra banda, i a efectes d'algorísmica i de programació, també és útil considerar el vector  $p$ -èsima columna de la matriu  $\mathbf{Z}$ :  $\mathbf{z} = \mathbf{w} + \mathbf{e}_p$ , on  $\mathbf{e}_p$  és el  $p$ -èsim vector canònic complex. L'única diferència entre els vectors  $\mathbf{w}$  i  $\mathbf{z}$  és la presència de l'element  $(0,0)$  o  $(1,0)$  en la posició  $p$ . En el text es farà referència al vector que permeti una escriptura més ràpida i còmoda en cada moment, però sempre es podran fer interpretacions en termes de l'altre.

L'estructura particular de la matriu de transformació  $Z$  utilitzada en el mètode APT la fa matemàticament i computacionalment molt manejable a causa de les següents propietats que es destaquen:

1. Només cal guardar en memòria un sol vector ( $w$  o  $z$ ).
2. El producte d'una matriu quadrada arbitrària  $A$  per la matriu  $Z$  té com a únic efecte la modificació del vector  $p$ -èsima columna de la matriu  $A$ : a aquesta columna se li ha addicionat el vector  $Aw$ .
3. Segons la propietat anterior, el producte de dues matrius amb l'estructura de  $Z$  també és una altra matriu amb la mateixa estructura. El vector  $w$  associat a la matriu resultant és la suma dels dos vectors  $w$  relatius a les dues matrius que es multipliquen. Això indica que aquest producte matricial és commutatiu.
4. La inversa de la matriu  $Z$  és simplement  $Z^{-1} = I - W$ , la qual té la mateixa estructura de  $Z$ .

Fent ús de les dues darreres propietats anteriors, es demostra fàcilment que la transformació de semblança que provoca la matriu  $Z$  definida a (2) sobre la matriu  $H$  que apareix a (1) es codifica a través del conjunt de transformacions següent:

- a) Per  $i=p, j=p : h_{pp} \leftarrow \sigma_p$
- b) Per  $i \neq p, j=p : h_{ip} \leftarrow \sigma_i - z_i \sigma_p$
- c) Per  $i=p, j \neq p : h_{pj} \leftarrow h_{pj}$
- d) Per  $i \neq p, j \neq p : h_{ij} \leftarrow h_{ij} - z_i h_{pj}$

on  $\{h_{ij}\}$  són els elements de la matriu  $H$ ,  $\{z_i\}$  són els elements del vector  $z$  i els termes  $\{\sigma_i\}$  són els elements del vector  $\sigma$ , el qual està definit com a

$$\sigma = Hz. \tag{5}$$

L'expressió (4) assenjala quina forma pren la matriu  $H$  just després de l'aplicació de la primera matriu de transformació  $Z$ . Els nous termes que apareixen a la columna  $p$  defineixen el vector residu o remanent que s'ha de diagonalitzar:

$$r = \left\{ r_i = \sigma_i - z_i \sigma_p \quad \forall i \neq p \quad ; \quad r_p = (0, 0) \right\} \tag{6}$$

i a partir d'ells s'obtindrà la nova transformació que s'ha d'aplicar.

Per tal de dirigir la transformació (4) cap a la diagonalització de la columna  $p$ -èsima de  $H$ , cal que el vector inicial  $z$  considerat sigui com més proper millor al vector propi buscat. Si no es disposa de més informació, el més comú és definir el vector  $w$  a partir de la correcció de primer ordre de la teoria de perturbacions de Rayleigh-Schrödinger. Llavors, en la primera iteració, el vector  $w$  esdevé:

$$w = \left\{ w_i = \frac{h_{ip}}{h_{pp} - h_{ii}} \quad \forall i \neq p \quad ; \quad w_p = (0, 0) \right\} \tag{7}$$

Per a les iteracions que segueixen també es pot aplicar la correcció pertorbativa de primer ordre. Per exemple, els següents vectors  $w$  que cal considerar són del tipus:

$$w = \left\{ w_i = \frac{r_p}{\sigma_p - h_{ii} + z_i h_{pi}} \quad \forall i \neq p \quad ; \quad w_p = (0,0) \right\} \quad (8)$$

els quals s'obtenen aplicant la mateixa recepta (7) sobre la nova matriu definida a (4).

Per tal d'obtenir un algorisme que no modifiqui la matriu original es treu profit de la propietat 3 descrita més amunt i associada a les matrius del tipus  $Z$ : no s'aplica una nova matriu de transformació  $Z$  sobre la matriu  $H_i$  de (1), sinó que s'avalua quina és la transformació global efectiva que s'aplicaria sobre la matriu original  $H$  si això es fes. En aquest cas cal considerar quin és el producte de dues transformacions del tipus  $Z$ . Segons la propietat 3, aquest producte origina una altra matriu del mateix tipus. Per obtenir-la, segons la mateixa propietat 3, només cal fer una suma vectorial sobre la columna rellevant de la matriu  $Z$  original. A efectes pràctics, això es tradueix a anar afegint termes sobre el vector  $z$  o  $w$  cada vegada que s'aplica una nova transformació de semblança. És aquesta característica la que permet concebre el procés de diagonalització de manera iterativa i sense haver de transformar la matriu original  $H$ . En resum, el que es fa a cada iteració és afegir el vector definit a (8) sobre el vector  $w$  previ i llavors es torna a aplicar la nova matriu  $Z$  resultant utilitzant de nou el producte (1), i així s'obté una nova matriu transformada del tipus (4), la qual permet calcular el nou vector  $w$  segons la fórmula (8). Quan els elements del vector residu definit a (6) es fan negligibles, les correccions pertorbatives que cal considerar també ho són i es pot considerar acabat el procediment de diagonalització.

Al final del procés iteratiu, el vector  $z$  esdevé el vector propi no normalitzat de  $H$ . La presència de l'element (1,0) a la posició  $p$  evidencia que, al llarg de tot el procés, s'ha estat treballant sota el marc d'una normalització intermèdia. A cada iteració, l'element  $\sigma_p$  que s'ha calculat constitueix una nova aproximació al valor propi buscat.

## COMENTARIS SOBRE EL MÈTODE

El mètode APT presenta diversos avantatges i inconvenients algebriacs o computacionals i també permet que li siguin implementades millores. A continuació es descriuen breument algunes d'aquestes característiques. Moltes s'estan analitzant actualment al nostre laboratori.

### *Avantatges del mètode*

- Només cal guardar en memòria un parell de vectors:  $z$  i  $\sigma$ .
- Es basa en l'aplicació d'una transformació de semblança general i això permet que es pugui diagonalitzar qualsevol tipus de matriu, ja sigui simètrica o no i tant si té elements reals com imaginaris.

- El mètode convergeix molt més ràpidament que la teoria de pertorbacions de Rayleigh-Schrödinger (1).
- Com que es tracta d'un mètode iteratiu, només la darrera iteració aporta l'error numèric d'acumulació.
- El seu cost és lineal en relació amb el nombre d'iteracions. Això fa que es puguin assolir ordres de correcció de tipus pertorbacional amb un cost molt més baix que l'ofert per la teoria de pertorbacions de Rayleigh-Schrödinger. Aquesta característica serà estudiada en un futur al nostre laboratori per tal de trobar expressions alternatives a les conegudes de Møller i Plesset (6) en el marc de la química quàntica.
- Com que no transforma directament la matriu  $\mathbf{H}$ , el mètode es pot utilitzar per diagonalitzar matrius de grans dimensions. La matriu que s'ha de diagonalitzar només intervé de manera global en l'avaluació del producte (5).

### *Característiques limitants del mètode*

- És aplicable a la diagonalització de qualsevol columna només quan aquesta té una diagonal prou dominant o s'aplica en la primera iteració un vector  $\mathbf{z}$  molt proper al vector propi cercat.
- Es pot considerar que el mètode APT només tracta un subespai monodimensional  $i$ , per tant, la seva convergència és lenta en comparació amb altres mètodes que treballen amb subespais de dimensió reduïda (7, 8, 9).

### *Algunes millores addicionals*

- En lloc d'afegir directament els nous vectors de correcció  $\mathbf{w}$ , es poden addicionar múltiples d'aquest sota el criteri de cercar una minimització del mòdul del vector residu (6) que es genera a cada iteració.
- Si es guarden les darreres correccions utilitzades, es pot treballar amb subespais de dimensió arbitrària, tal com fan altres mètodes, com per exemple el de Davidson (10) per a matrius simètriques. D'aquesta manera es pot concebre el procediment APT com un generador de vectors pels mètodes de diagonalització per aproximació des d'un subespai (11).
- En la primera iteració es pot utilitzar una bona aproximació al vector propi cercat. Per exemple, aquesta es pot obtenir utilitzant correccions d'ordre superior al primer de la teoria de pertorbacions o bé prenent el vector que sorgeix de la diagonalització d'una submatriu de  $\mathbf{H}$ .
- El mètode es pot adaptar per calcular diversos vectors propis alhora (11). Si això es posa en pràctica, a cada iteració es poden depurar les actuals aproximacions als vectors propis per tal que aquestes siguin ortogonals als vectors propis que ja hagin convergit.
- D'altra banda, el coneixement del vector  $\sigma$  definit a (5) permet calcular fàcilment una altra aproximació al valor propi a través del quocient de Rayleigh-Ritz.
- Per tal d'agilitar els càlculs en l'avaluació de les correccions descrites a (8) es poden guardar en memòria un o dos vectors addicionals: els que contenen la diagonal i la *filera*  $p$ -èsima de la matriu original  $\mathbf{H}$ .

## ALGORISME I CODI COMPUTACIONALS

L'algorisme 1 constitueix la codificació del mètode que s'acaba de descriure. En cas de convergència, el vector  $z$  acaba emmagatzemant un vector propi no normalitzat de la matriu  $H$  i la variable  $e$  és el valor propi associat. El criteri de convergència està governat pel paràmetre real  $\epsilon$ : quan tots els components del vector residu (6) tenen mòdul inferior a aquesta quantitat, el procés s'atura i es considera que ha convergit.

1. Inicialitzacions:  $k = 0$ . Comptador d'iteracions.  
 $k_{max} > 0$ . Nombre màxim d'iteracions permès.  
 $1 \leq p \leq n$ . Columna per diagonalitzar.  
 $\epsilon > 0$ . Tolerància del procediment.  
 $\delta = \infty$ . Test de convergència.
2. Vector inicial obtingut a partir de les correccions perturbatives de primer ordre:

$$z = \left\{ z_i = \frac{h_{ip}}{h_{pp} - h_{ii}} \quad \forall i \neq p; z_p = (1, 0) \right\}$$

3. Mentrestant s'ha de fer  $\delta > \epsilon$  i  $k < k_{max}$ 
  - a)  $\sigma = Hz$ : coll d'ampolla del mètode.
  - b)  $e = \sigma_p$ : aproximació actual al valor propi.
  - c) Afegeix correcció de primer ordre al vector propi.  
S'obté la nova aproximació millorada al vector propi.  
Fem  $\forall i \neq p$ :

$$z_i \leftarrow z_i + \frac{\sigma_i - z_i e}{e - h_{ii} + z_i h_{pi}}$$

- d)  $\delta = \max_i \{ |\sigma_i - z_i e| \}$ : Màxim mòdul en els components del residu.
- e)  $k \leftarrow k + 1$ : incrementa el comptador d'iteracions.

4. Si  $\delta > \epsilon$  el procediment no ha convergit. S'atura el procés.

5. Convergència assolida. Fi.

**Algorisme 1.** Troba un valor propi ( $e$ ) i el corresponent vector propi no normalitzat ( $z$ ) d'una matriu quadrada  $H = \{h_{ij}\}$  d'ordre  $n$  amb elements pertanyents al cos complex. Es diagonalitza la  $p$ -èsima columna. Els vectors de correcció que requereix el mètode APT s'obtenen a partir de la correcció de primer ordre de la teoria de perturbacions de Rayleigh-Schrödinger.

El **programa 1** és el codi en Fortran 90 que implementa l'**algorisme 1**. En la seva confecció s'ha procurat utilitzar els mateixos noms de variables que els que apareixen al text i a l'**algorisme 1** per tal de facilitar-ne la lectura i interpretació.

```

!-----
logical function APT_Complex(H,epsilon,kmax,z,e,n,p,delta)
!-----
! Implementacio de l'algorisme APT.
! La funció retorna .true. si ha convergit satisfactòriament.
! Es canvia el valor de kmax.
!
! Entrada:
!   n : ordre de la matriu
!   H(n,n) : matriu per diagonalitzar.
!   epsilon : tolerància del procés.
!   kmax : nombre màxim d'iteracions que se suportarà.
!   p : índex de la columna per diagonalitzar.
!
! Sortida:
!   e : valor propi trobat.
!   z(n) : vector propi no normalitzat corresponent.
!   delta : mòdul màxim en els components del vector residu.
!   kmax : nombre d'iteracions que s'ha efectuat.
!-----
IMPLICIT DOUBLE PRECISION (a-h,o-z)
INTEGER p
COMPLEX*16 H(n,n),z(n),sigma(n),e,Ri

z(p)=(1.0d0,0.0d0) ! Vector z inicial
DO i=1,n; IF (i/=p) z(i)=H(i,p)/(H(p,p)-H(i,i)); END DO

k=0 ! Comptador d'iteracions
delta=epsilon+1.0d0
DO WHILE (delta>epsilon.and.k<kmax) ! Bucle iteratiu
  k=k+1
  sigma=MATMUL(H,z) ! Coll d'ampolla del mètode
  e=sigma(p) ! Nova aproximacio al valor propi
  delta=0.0d0 ! Màxim mòdul dels components del residu
  DO i=1,n
    IF (i/=p) THEN
      Ri=sigma(i)-z(i)*e ! Component del vector residu
      z(i)=z(i)+Ri/(e-H(i,i)+z(i)*H(p,i))
      delta=MAX(delta,REAL(Ri)**2+AIMAG(Ri)**2)
    END IF
  END DO
  delta=SQRT(delta)
END DO
IF (delta>epsilon) THEN
  APT_Complex=.false. ! No convergeix
ELSE
  APT_Complex=.true.; kmax=k ! Ha convergit!
END IF
END
!-----

```

*Programa 1. Codificació en Fortran 90 de l'algorisme 1. En aquesta versió la matriu diagonalitzar s'ubica a la memòria principal de l'ordinador.*

El coll d'ampolla de l'algorisme és el producte matriuvector  $\sigma = \mathbf{H}z$ . Aquest està codificat a través d'una funció pròpia del llenguatge Fortran 90 i és per això que el codi que es presenta requereix encabir tota la matriu en la memòria de l'ordinador. És clar que quan les dimensions de la matriu no permeten disposar de tota la matriu emmagatzemada en memòria, caldrà codificar el producte a través d'una funció definida per l'usuari. En aquests casos, el més típic és generar el vector  $s$  resultant a través de les diferents contribucions dels elements de la matriu  $\mathbf{H}$ . Tanmateix, aquest procediment sempre depèn directament del tipus d'aplicació concreta que origina la matriu que s'ha de diagonalitzar.

Tant l'algorisme com el codi de programació treballen de manera explícita amb el vector  $z$ . Tot i això, les redefinicions que es fan en el pas 3c de l'algorisme 1 són addicions del nou vector  $w$  sobre el vector  $z$  antic. Aquesta addició no és res més que el càlcul encobert del producte de dues matrius del mateix tipus que  $\mathbf{Z}$ , tal com s'assenyala en els comentaris de la propietat 3 d'aquestes matrius.

### EXEMPLES NUMÈRICS

Com a exemple d'aplicació del mètode descrit s'han diagonalitzat algunes columnes de diverses matrius quadrades no simètriques d'ordre  $n$  i de variable complexa. Aquestes matrius tenen els seus elements definits com a:

$$\mathbf{H} = \left\{ h_{KL} = \frac{1}{\gamma_{KL}} \frac{1}{(K-L)} \quad \forall K, L = 1, 2, \dots, n \right\} \quad (9)$$

on  $(K,L)$  és el nombre complex  $K+iL$  i el paràmetre  $\gamma_{KL}$  és real. Aquest darrer factor s'ha introduït per tal que la matriu tingui una diagonal més dominant i que asseguri la convergència del mètode. De fet, el valor de  $\gamma_{KL}$  només afecta els elements no diagonals de la matriu  $\mathbf{H}$  perquè la seva definició és:

$$\gamma_{KL} = \begin{cases} 1 & \text{si } K = L \\ \gamma & \text{si } K \neq L \end{cases} \quad (10)$$

essent  $|\gamma| > 1$ .

Els resultats obtinguts es poden consultar a les taules 1 i 2, les quals són autoexplicatives.



		Columna diagonalitzada		
		1	2	3
Iteracions		13	22	30
Valor propi		(0.5112474044,-0.5112474044)	(0.2632789713,-0.2632789721)	(0.1811093020,-0.1811093032)
Primeres components del vector propi	1:	(1,0,0)	(-0.21637668,0.055847916)	(-0.099983578,0.021348981)
	2:	(0.13843356,0.04267862)	(1,0,0)	(-0.69038057,0.092336006)
	3:	(0.077475957,0.036401212)	(0.29062457,0.051455340)	(1,0,0)
	4:	(0.053697777,0.030861455)	(0.18591493,0.056407756)	(0.43744632,0.051649864)
	5:	(0.041003259,0.026705480)	(0.13933804,0.055800227)	(0.30928119,0.067003585)
MMCR <sup>1</sup>		1.51·10 <sup>9</sup>	4.02·10 <sup>9</sup>	3.01·10 <sup>9</sup>
MVR <sup>2</sup>		4.66·10 <sup>9</sup>	1.26·10 <sup>8</sup>	7.13·10 <sup>9</sup>

<sup>1</sup> Màxim mòdul trobat en els components del vector residu després de la convergència.  
<sup>2</sup> Mòdul del vector residu després de la convergència.

**Taula 1.** Resultats obtinguts en el càlcul dels tres primers vectors i valors propis de mòdul més gran d'una matriu d'ordre 100 amb els elements descrits a l'equació (9). Els elements no diagonals han estat escalats en un factor  $\gamma=10$ . El paràmetre de tolerància utilitzat a l'algorisme 1 ha estat  $\epsilon=10^{-8}$ .

Ordre de la matriu	Coefficient $\gamma$	Nombre d'iteracions	Temps (s) <sup>1</sup>	Valor propi trobat	MMCR <sup>2</sup>	MVR <sup>3</sup>
10	1.0	11	-	(1.194105047,-1.194105045)	6.14·10 <sup>9</sup>	1.31·10 <sup>8</sup>
	10.0	9	-	(0.5091185738,-0.5091185738)	1.64·10 <sup>9</sup>	2.99·10 <sup>9</sup>
	100.0	4	-	(0.5000788169,-0.5000788169)	1.57·10 <sup>11</sup>	2.8·10 <sup>11</sup>
100	10.0	13	-	(0.5112474044,-0.5112474044)	1.51·10 <sup>9</sup>	4.66·10 <sup>9</sup>
	100.0	4	-	(0.5000885948,-0.5000885948)	1.02·10 <sup>10</sup>	3.02·10 <sup>10</sup>
1000	10.0	14	35	(0.5116511200,-0.5116511198)	4.72·10 <sup>9</sup>	2.05·10 <sup>8</sup>
	100.0	4	10	(0.5000896294,-0.5000896294)	1.54·10 <sup>10</sup>	6.09·10 <sup>10</sup>
10000	100.0	4	993	(0.5000897379,-0.5000897379)	1.69·10 <sup>10</sup>	7.91·10 <sup>16</sup>
	500.0	3	745	(0.5000035149,-0.5000035149)	1.38·10 <sup>12</sup>	6.03·10 <sup>12</sup>
100000	1000.0	2	50459	(0.5000008765,-0.5000008765)	1.14·10 <sup>11</sup>	4.61·10 <sup>11</sup>

<sup>1</sup> Càlculs efectuats en un ordinador PC Pentium de 200 MHz. No inclou el temps de càlcul del mòdul del vector residu.  
Per a matrius de dimensió 100 o menor aquest temps és negligible.  
<sup>2</sup> Màxim mòdul trobat en els components del vector residu després de la convergència.  
<sup>3</sup> Mòdul del vector residu després de la convergència.

**Taula 2.** Resultats obtinguts en el càlcul del valor propi de mòdul més gran de matrius del tipus (9) de diversos ordres. Els elements no diagonals han estat escalats per un valor del paràmetre  $\gamma$  específic per a cada matriu. La tolerància utilitzada a l'algorisme 1 ha estat  $\epsilon=10^{-8}$ .

## CONCLUSIONS

S'ha descrit el procediment iteratiu APT, el qual permet trobar valors i vectors propis de matrius arbitràries amb diagonal prou dominant. S'han donat tant l'algorisme del mètode com un programa en el llenguatge Fortran 90. Finalment, s'han mostrat alguns exemples d'aplicació numèrica.

## AGRAÏMENTS

Aquest treball ha estat finançat per la CICYT en el Programa Nacional de Salud y Farmacia a través del projecte SAF96-6158.

## Bibliografia

1. BESALÚ, E. i CARBÓ-DORCA, R., *J. Math. Chem.*, 1997, 21,395-412.
2. FADDEEVA, V.N., "Computational Methods of Linear Algebra". Dover Pub. Inc., Nova York, 1959.
3. a) WILKINSON, J.H., "The Algebraic Eigenvalue Problem". Clarendon Press, Oxford, 1965.  
b) WILKINSON, J.H. i REINSCH, C., "Linear Algebra". Springer-Verlag. Berlín, 1971.
4. GOLUB, G.H. i VAN LOAN, C.H.F., "Matrix Computations". The Johns Hopkins University Press, Londres, 1993.
5. DURAND, E., "Solutions Numériques des équations Algébriques". Vol II. Masson & Cie., París, 1961.
6. MØLLER, C. i PLESSET, M.S., *Phys. Rev.*, 1934, 46, 618.
7. ARNOLDI, W.E., *Quart. Appl. Math.*, 1951, 9, 17-29.
8. LANCZOS, C., *J. Res. Nat. Bur. Standards*, 1950, 45, 255-282.
9. KARUSH, W., *Pacific J. Math.*, 1951, 1, 233-248.
10. a) DAVIDSON, E.R., *J. Comp. Phys.*, 1975, 17, 87-94.  
b) DAVIDSON, E.R., *J. Phys.*, 1980, A13, L179.
11. BESALÚ, E. i BOFILL, J.M., *J. Comp. Chem.*, (enviat).