DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN
UNIVERSIDAD POLITÉCNICA DE VALENCIA

P.O. Box: 22012 E-46071 Valencia (SPAIN)

Informe Técnico / Technical Report

Ref. No.: DSIC-II/15/05 **Pages: 15**
Title: *Estudio de la Eficiencia de Librerías Numéricas de Libre Distribución (PETSc, SPARSKIT) a la Resolución de los Sistemas de Ecuaciones Lineales relacionados con la Ecuación de los Modos Lambda*
Author(s): *O. Flores-Sánchez*
Date: *1 de Noviembre, 2005*
Keywords: *Métodos del subespacio de Krylov, Sistemas Lineales Dispersos, Métodos Iterativos, Precondicionadores, Computación Paralela y Distribuida*

Vº Bº
Leader of research Group
Vicente Hernández García

Author(s)
O. Flores-Sánchez

Estudio de la Eficiencia de Librerías Numéricas de Libre Distribución (PETSc, SPARSKIT) a la Resolución de los Sistemas Lineales Dispersos relacionados con la Ecuación de los Modos Lambda

Omar Flores Sánchez

REPORTE TÉCNICO
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN
Universidad Politécnica de Valencia

Índice

1. Introducción

La modelización matemática de muchos fenómenos físicos involucra la utilización de ecuaciones en derivadas parciales (EDPs). Uno de los fenómenos físicos que se busca modelar es la difusión de neutrones en el interior del núcleo de un reactor, donde se precisa la resolución numérica rápida y eficiente de la *Ecuación de Difusión Neutrónica* (EDN) para su simulación.

De la Ecuación de Difusión Neutrónica se derivan dos cálculos distintos, aunque complementarios. Un primer tipo de cálculo es el que determina la configuración estática del reactor en un instante de tiempo dado, y que toma la forma de un problema de *valores propios generalizado*. El otro tipo de cálculo se realiza para el estudio de un transitorio a partir de una perturbación efectuada sobre una configuración estática del reactor, utilizando para ello la Ecuación de la Difusión Neutrónica en su forma dependiente del Tiempo.

La forma común de resolver la EDN es discretizándola, es decir, aproximando las Ecuaciones Diferenciales Parciales mediante ecuaciones algebraicas que involucren un número finito de incógnitas. La utilización de métodos de discretización permite reducir el problema original a la resolución de un problema algebraico de sistemas de ecuaciones lineales cuya matriz de coeficientes es generalmente dispersa y de gran dimensión.

Dentro de los métodos para resolver numéricamente los sistemas de ecuaciones lineales resultado de la discretización de las EDPs se encuentran los métodos *directos* y los métodos *iterativos*. Los métodos directos hasta hace poco se habían estado utilizando preferentemente sobre los métodos iterativos dada su robustez y comportamiento predecible; sin embargo, los métodos iterativos han demostrado buena competencia dada la aparición de técnicas de precondicionado, que combinadas con iteraciones sobre *subespacios de Krylov* han proporcionado procedimientos de propósito general eficientes y sencillos. Los métodos iterativos tienen también la particularidad de que pueden implementarse sobre computadores de alto desempeño más fácilmente que los métodos directos[1].

Así pues, se hace necesario estudiar, aplicar, desarrollar métodos computacionales que permitan reducir los costes asociados a la resolución de sistemas lineales dispersos de gran dimensión como los que aparecen en la simulación de fenómenos físicos.

2. Ecuación de los Modos Lambda

Para estudiar la distribución de flujo neutrónico en estado estacionario del núcleo de un reactor de potencia nuclear, y los modos subcríticos responsables de las inestabilidades regionales producidas en los reactores, es necesario obtener los λ-autovalores y sus correspondientes autovectores asociados con un sistema de ecuaciones diferenciales parciales de la forma

$$-\overrightarrow{\nabla}\cdot D_g(\overrightarrow{r})\overrightarrow{\nabla}\phi_g(\overrightarrow{r})+\sum_{g}^{r}\phi_g(\overrightarrow{r})-\sum_{g\neq g'}^{G}\sum_{gg'}^{s}\phi_{g'}(\overrightarrow{r})=\frac{1}{\lambda}x_g\sum_{g'=1}^{G}v\sum_{g'}^{f}\phi_{g'}(\overrightarrow{r}) \quad (1)$$

con $g=1,2,\ldots,G$, donde G representa el número de grupos de energía. La condición de frontera para el problema es $\phi_g|_\Gamma=0$, en la que Γ es el contorno del reactor. Este problema es conocido como la EDN *estática multigrupo* [2].

Si la ecuación (1) es modelada con dos grupos de energía (un grupo *térmico* (ϕ_t) y otro *rápido*(ϕ_f)), entonces el problema a tratar es encontrar los autovalores y autofunciones de

$$\mathcal{L}\phi_i=\frac{1}{\lambda_i}\mathcal{M}\phi_i, \quad (2)$$

la cual se conoce como ecuación de los *modos lambda*, en donde

$$\mathcal{L}=\begin{bmatrix}-\overrightarrow{\nabla}\cdot(D_1\overrightarrow{\nabla})+\sum_{a1}+\sum_{12} & 0\\ -\sum_{12} & -\overrightarrow{\nabla}\cdot(D_2\overrightarrow{\nabla})+\sum_{a2}\end{bmatrix}, \quad (3)$$

$$\mathcal{M}=\begin{bmatrix}v_1\sum_{f1} & v_2\sum_{f2}\\ 0 & 0\end{bmatrix}, \qquad \text{y} \qquad \phi_i=\begin{bmatrix}\phi_{f_i}\\ \phi_{t_i}\end{bmatrix} \quad (4)$$

Tras discretizar la ecuación e imponer las condiciones de continuidad y de contorno adecuadas, éste se transforma en el siguiente problema algebraico de valores propios generalizado

$$L\psi_n=\frac{1}{k_n}M\psi_n \quad (5)$$

donde L y M son matrices de dimensión $2N$, con la siguiente estructura a bloques N-dimensional.

$$\begin{bmatrix}L_{11} & 0\\ -L_{21} & L_{22}\end{bmatrix}\begin{bmatrix}\psi_{1_n}\\ \psi_{2_n}\end{bmatrix}=\frac{1}{k_n}\begin{bmatrix}M_{11} & M_{12}\\ 0 & 0\end{bmatrix}\begin{bmatrix}\psi_{1_n}\\ \psi_{2_n}\end{bmatrix} \quad (6)$$

Dependiendo de las condiciones de continuidad del flujo que se imponen entre las celdas en que se ha discretizado el núcleo del reactor, las matrices

L_{11} y L_{22} serán *simétricas* o no. Debido a las propiedades físicas las matrices son *diagonal dominantes* y *definidas positivas*, por lo que la convergencia está garantizada. Por otra parte, bajo ciertas condiciones, el bloque nulo del operador $\mathcal{L}$ puede ser una matriz diagonal como lo son L_{21}, M_{11} ó M_{12}. Todas las matrices son *dispersas* y de *gran dimensión*.

Uno de los enfoques utilizados para la resolución del problema de valores propios generalizado de dimensión $2N$ representado en (6), es reducirlo a un problema ordinario de dimensión N. Así pues, de la ecuación (6) se deducen las dos siguientes,

$$L_{11}\psi_{1_n} = \frac{1}{k_n}(M_{11}\psi_{1_n} + M_{12}\psi_{2_n}), \tag{7}$$

$$-L_{21}\psi_{1_n} + L_{22}\psi_{2_n} = 0. \tag{8}$$

Si se despeja ψ_{2_n} en la ecuación (8) y se sustituye su valor en (7), se obtiene la siguiente expresión

$$S\psi_{1_n} = k_n\psi_{1_n}, \tag{9}$$

donde la matriz S viene dada por

$$S = L_{11}^{-1}(M_{11} + M_{12}L_{22}^{-1}L_{21}).$$

En métodos de cálculo de autovalores basados en el *Método de Arnoldi*, como se describe en el Algoritmo 1, la operación más costosa, desde el punto de vista computacional, está representada por operaciones tipo *matriz dispersa-vector* (Paso 4); sin embargo, dado que no contamos con la matriz S en forma explícita, es necesario realizar la serie de pasos mostrados en el Algoritmo 2, y que muestra tres operaciones *matriz diagonal-vector*, una *suma* de vectores, y la *resolución* de los sistemas de ecuaciones lineales dispersos con L_{22} y L_{11}. De dichas operaciones, las relacionadas con los sistemas de ecuaciones lineales son las más costosas, por lo que su resolución eficiente, significaría una mejora proporcional en el proceso global.

Esta estrategia ha sido desarrollada con éxito en métodos tales como *Iteración del Subespacio* o en variantes del método de Arnoldi como el *Método de Arnoldi con Reinicio Implícito* [3].

3. Caso de Estudio

El reactor de la central nuclear sueca de Ringhals I, de tipo agua en ebullición (BWR), se ha discretizado de forma tridimensional en 27 planos

1	**Inicio** : *Elegir un vector* v_1 *de norma 1.*
2	**Iterar** : *for* $j = 1, 2, \ldots, m$ *calcular:*
3	$h_{i,j} = (Sv_j, vi)$, $i = 1, 2, \ldots, j$,
4	$w_j = Sv_j - \sum_{i=1}^{j} h_{ij} v_i$,
5	$h_{j+1,j} = \parallel w_j \parallel_2$, if $h_{j+1,j} = 0$ *parar.*
6	$v_{j+1} = w_j / h_{j+1,j}$.

Algoritmo 1: Método de Arnoldi.

1	$w_1 = L_{21} v$
2	$w_2 = L_{22}^{-1} w_1$
3	$w_3 = M_{12} w_2$
4	$w_4 = M_{11} v$
5	$r = L_{11}^{-1}(w_3 + w_4)$

Algoritmo 2: Cálculo del producto Sv.

axiales de 14.72 cm de longitud, 25 correspondientes al combustible, un plano superior y otro inferior correspondientes al reflector. A su vez, cada uno de los planos axiales se divide en celdas de 15.275 cm por 15.275 cm. Cada una de las celdas tiene propiedades neutrónicas distintas [4]. Utilizando un esquema de numeración natural para la malla de discretización, el patrón de elementos no nulos resultante es de tipo banda con una dimensión de n =78624, con nnz =472625 y nnz =472223 elementos no nulos respectivamente (Véase Figura 1).

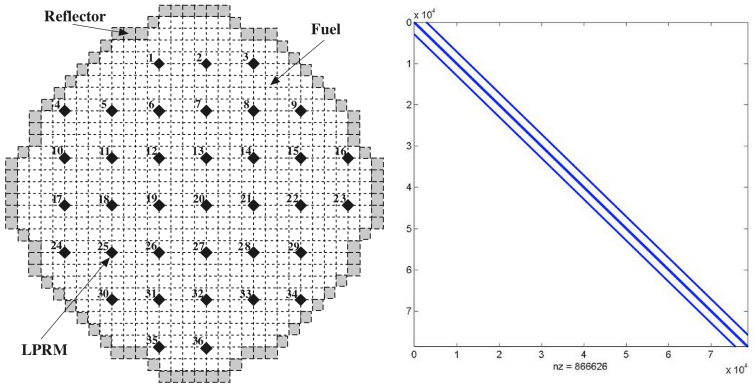

Figura 1: Esquema de discretización y Bloque L_{11} del caso Ringhals.

4. Plataforma Hardware y Software

Los experimentos fueron hechos en uno de los clusters del GRYCAP (Grupo de Redes y Computación de Altas Prestaciones) de la Universidad Politécnica de Valencia. Dicho cluster consta de 20 nodos biprocesadores Pentium Xeon a 2.0 Ghz, interconectados mediante una red SCI con topología Toro 2D en malla 4x5. Cada nodo tiene 1 Gigabyte de memoria RAM. Para las pruebas realizadas en este reporte, no se habilitó la característica de biprocesamiento.

Se ha utilizado la librería numérica paralela de PETSc (Portable, Extensible Toolkit for Scientific Computation) [5] [6] [7] [8] en su versión 2.2.0. Por otro lado, se ha utilizado la librería SPARSKIT [9] en su versión 2.0, el cual es un paquete de subrutinas FORTRAN para la manipulación de matrices dispersas, operaciones básicas del álgebra lineal dispersa, rutinas de conversión de formatos, generación de matrices, etc.

Los métodos iterativos basados en subespacios de Krylov que se usaron, fueron los que se encuentran en ambas librerías y se recogen en la Tabla 1.

Método	Solver
Gradiente Conjugado	CG
Gradiente Bi-Conjugado	BCG
BCG Estabilizado.	BCGSTAB
Residuo Quasi-Mínimo Libre Transpuesto.	TFQMR
Residuo Mínimo Generalizado.	GMRES

Tabla 1: Métodos iterativos contenidos en SPARSKIT y PETSc.

Descripción	Precondicionador
Factorización ILU con estrategia de truncamiento dual.	ILUT
Precondicionamiento ILU(0) simple.	ILU0
Precondicionador de Jacobi	JAC

Tabla 2: Precondicionadores de SPARSKIT

Los métodos iterativos, suelen proporcionar mejores resultados cuando se combinan con técnicas de precondicionado. SPARSKIT tiene implementados varios precondicionadores secuenciales, de los cuales se probaron en este trabajo los contenidos en la Tabla 2; por su lado PETSc proporciona los precondicionadores de la Tabla 3.

5. Experimentos Numéricos

Los cálculos numéricos tanto en SPARSKIT como PETSc, se hicieron utilizando aritmética de punto flotante con doble precisión. En los experimentos se ha elegido un test de convergencia basado en la l_2-norma del residuo, por lo que la convergencia se detecta en la iteración k si

$$\|r_k\|_2 < \epsilon * \|b\|_2,$$

Precondicionador	Opción PETSc
Jacobi	PCJACOBI
Jacobi a Bloques	PCBJACOBI
Método Aditivo de Schwarz	PCASM

Tabla 3: Precondicionadores probados de PETSc

Métodos	L_{22}		L_{11}		T_t
	Its.	T_1	**Its.**	T_1	
GC	59	0.58	121	1.20	**1.78**
BCG	116	1.16	240	2.44	3.60
GMRES	70	1.22	169	2.99	4.21
BCGSTAB	83	0.80	155	1.52	2.32
TFQMR	65	0.74	149	1.70	2.44

Tabla 4: Tiempos de los métodos de SPARSKIT (Sin Precondicionar)

donde $r_k = b - Ax_k$. La tolerancia exigida para los distintos métodos fué de $\epsilon = 10^{-12}$, y se eligió como punto inicial $x_0 = \mathbf{0}$, por lo que $r_0 = b$. El número máximo de iteraciones ($maxits$) se estableció a 10000.

Para poder contrastar más adelante las ganancias obtenidas en el desempeño de los métodos iterativos basados en subespacios de Krylov y determinar el menor tiempo secuencial, se realizaron pruebas de ejecución sin precondicionador, obteniéndose los tiempos de la Tabla 4, en la que se observa que el costo en tiempo de ejecución para resolver el sistema L_{22} es menor que el invertido para resolver el sistema L_{11} en todos los métodos. Se observa también que el método más rápido es el *Gradiente Conjugado* (CG) al resolver los dos sistemas; y el menos rápido ha sido el *Residuo Mínimo Generalizado* (GMRES). En la tabla, los símbolos T_1 y T_t hacen referencia al *tiempo secuencial* y a la suma de los tiempos secuenciales invertidos respectivamente, en resolver los sistemas L_{22} y L_{11}.

Uno de los precondicionadores más sencillos es el precondicionador *diagonal* o de *Jacobi*. En esta técnica, el precondicionador es la diagonal principal de la matriz de coeficientes, por lo que su cálculo es rápido. Los tiempos registrados por el uso de este precondicionador combinada con los métodos iterativos del subespacio de Krylov se muestran en la Tabla 5. Como se observa, se ha obtenido una disminución en el número de iteraciones y por lo tanto en el tiempo para alcanzar la tolerancia de convergencia respecto a los obtenidos sin el uso de precondicionador. El método del CG sigue manteniéndose a la cabeza en la solución de ambos sistemas de ecuaciones lineales registrando un tiempo de 1.25 segundos, lo que significa una disminución en el tiempo respecto al caso sin precondicionar de un 30 %.

Al emplear una técnica de precondicionado tipo ILU0, se obtiene una rapidez de convergencia mayor como se observa en la Tabla 6, pues de 36 iteraciones con precondicionador de Jacobi se pasa a 14 lo que representa una reducción en el número de iteraciones del 61 %.

Métodos	L_{22}		L_{11}		T_t
	Its.	T_1	**Its.**	T_1	
GC	36	0.45	65	0.80	**1.25**
BCG	70	0.87	128	1.59	2.46
GMRES	41	0.82	81	1.60	2.42
BCGSTAB	49	0.60	95	1.15	1.75
TFQMR	39	0.53	75	1.01	1.54

Tabla 5: Tiempos de los métodos + PC de Jacobi en SPARSKIT

Métodos	L_{22}		L_{11}		T_t
	Its.	T_1	**Its.**	T_1	
GC	14	**0.39**	64	1.62	2.01
BCG	26	0.64	46	1.10	1.74
GMRES	15	0.54	27	0.89	1.43
BCGSTAB	15	0.42	29	**0.75**	1.17
TFQMR	15	0.43	29	0.80	1.23

Tabla 6: Tiempos de los métodos + PC ILU0 en SPARSKIT

Se aplicó un precondicionador tipo ILUT con distintos niveles de llenado (*lfil=1*), y tolerancia de caida ($DropTol$=1.0^{-2}), sin embargo, los resultados no fueron competitivos por lo que no se muestran aquí.

Basándonos en los resultados de la Tabla 6, el mejor tiempo secuencial está determinado por la suma del mejor tiempo en resolver el sistema L_{22} más el mejor tiempo en resolver el sistema L_{11} (Véase Tabla 7).

La **Figura 2** muestra la gráfica de convergencia de la mejor combinación de método y precondicionador aplicado a las matrices L_{22} y L_{11} en SPARSKIT y PETSc (con p=10), donde se observa que la convergencia para el sistema L_{22} es más rápida que en L_{11}.

De los experimentos paralelos en PETSc , la combinación más rápida la

	CG-ILU0	**BCGS-ILU0**	
	L_{22}	L_{11}	T_1(segs.)
Solución	**0.39**	**0.7**	**1.14**

Tabla 7: Mejor tiempo secuencial (T_1) en SPARSKIT

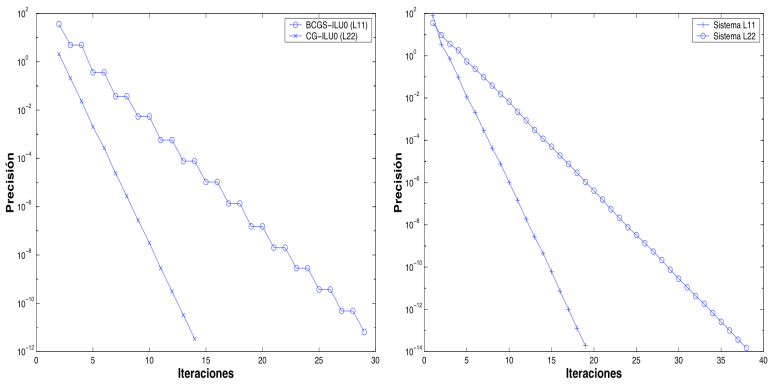

Figura 2: Convergencia de los sistemas L_{22} y L_{11} en SPARSKIT y PETSc respectivamente.

Método	$p = 2$	$p = 4$	$p = 8$	$p = 10$
CG	0.80	0.43	0.25	**0.22**
BICG	1.51	0.81	0.45	0.38
GMRES	1.07	0.58	0.30	0.26
BCGS	0.91	0.49	0.29	0.23
TFQMR	0.90	0.49	0.26	0.23

Tabla 8: Tiempos de solución en PETSc con precondicionador Jacobi a bloques.

proporciona el precondicionador de tipo Jacobi a Bloques (**BJacobi**) con el método del *Gradiente Conjugado* (CG), y cuyos resultados aplicados a las matrices L_{22} y L_{11} se muestran en la Tabla 8 para distintas configuraciones de procesadores (p).

Para determinar la ganancia de velocidad (*speedup*) obtenida por el programa paralelo sobre el secuencial, se calcula la razón del tiempo invertido para resolver el problema sobre un sólo procesador (T_1) al tiempo requerido en resolver el mismo problema sobre una computadora paralela con p procesadores idénticos (T_p), y se denota dicha razón por el símbolo S.

$$S = \frac{T_s}{T_p}$$

Otro parámetro de desempeño es la *eficiencia*, que es una medida de la fracción de tiempo para el cual un procesador es útilmente empleado y se define como la razón del *speedup* (S) respecto al número de procesadores p,

p	T_p	S	E
1	**1.14**	1.00	100.00 %
2	0.80	1.43	71.70 %
4	0.43	2.67	66.77 %
8	0.25	4.52	56.48 %
10	0.22	5.12	51.21 %

Tabla 9: Tiempos de solución en PETSc con precondicionador Jacobi a bloques.

y se denota por el símbolo E [10].

$$E = \frac{S}{p}$$

Para el caso de los sistemas de ecuaciones que estamos resolviendo, se han obtenido los coeficientes de *speedup* y *eficiencia* contenidos en la **Tabla 9**, donde el mejor tiempo secuencial está determinado por T_S =1.14 segundos (Véase Tabla 7).

La Tabla 9 muestra los resultados experimentales obtenidos de resolver los sistemas L_{22} y L_{11} con distintas configuraciones de procesadores, donde se observa la ventaja de utilizar computación de altas prestaciones (*High Performance Computing*) en la resolución de problemas de gran dimensión. Por ejemplo, de un tiempo secuencial de 1.14 segundos, pasamos a resolver los sistemas en 0.8 segundos al utilizar $p = 2$ procesadores, lo que representa una reducción en el tiempo secuencial del 30 % en cada producto matriz-vector, hecho que influye directamente en el tiempo del proceso global de cálculo de valores propios establecido inicialmente.

La **Figura 3** muestra la gráfica de *speedup*[10] obtenida con $p = 1, 2, 4, 8, 10$ procesadores, en donde se observa que al usar $p = 8$ procesadores hemos partido el tiempo secuencial hasta por un factor de 4.52, lo que significa una reducción sobre el tiempo secuencial de hasta un 78 %.

La **Figura 3** muestra que la *eficiencia*[10] paralela obtenida con distintos número de procesadores, es aceptable; sin embargo, si se desea mantener una eficiencia por arriba del 50 %, es necesario utilizar una configuración de $p \leq 10$ procesadores.

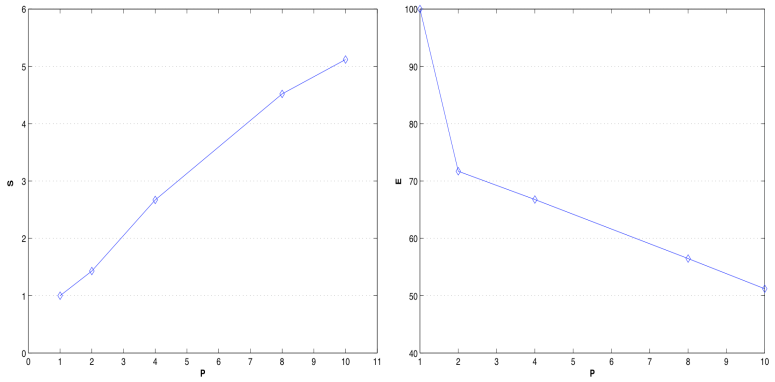

Figura 3: Aceleración (*speedup*) y Eficiencia para distintos valores de p.

6. Conclusiones

Hemos evaluado dos librerías numéricas para resolver sistemas de ecuaciones lineales dispersos: una secuencial y otra paralela. En la librería secuencial (SPARSKIT) se ha encontrado que el método más rápido resulta ser el CG (*Gradiente Conjugado*) para resolver el sistema L_{22}, y el método BCGSTAB (*Gradiente Bi-Conjuado Estabilizado*) para resolver el sistema L_{11}, ambos combinados con el precondicionador de tipo ILU0. En la librería paralela (PETSc), la combinación de método-precondicionador que mejores prestaciones obtuvo fué *Gradiente Conjugado* combinado con precondicionador tipo *Jacobi a Bloques*.

Las prestaciones secuenciales y paralelas obtenidas en los experimentos numéricos son aceptables, para el caso de estudio de los sistemas de ecuaciones lineales dispersos relacionados con la Ecuación de Difusión Neutrónica, y la ganancia de tiempo en la solución de dichos sistemas se verá reflejado en una reducción del tiempo de resolución del problema $Ax = \lambda x$.

No obstante, a pesar de obtener buenos parámetros de desempeño, una continuación al presente trabajo es la implementación tanto secuencial como paralela de métodos iterativos y precondicionadores que combinen técnicas de *dos etapas* [11], para buscar nuevas alternativas que permitan resolver rápida y eficientemente dichos sistemas de ecuaciones lineales asociados a cálculos tanto estáticos como dinámicos de la Ecuación de Difusión Neutrónica.

Referencias

[1] Saad Y. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, MA, 1996.

[2] Weston J.R. and Stacey M. *Space-Time Nuclear Reactor Kinetics*. Academic Press, 1969.

[3] Hernández V., Román J.E., Vidal A.M., and Vidal V. Calculation of lambda modes of a nuclear reactor: a parallel implementation using the implicitly restarted arnoldi method. In Springer, editor, *VECPAR'98 - 3rd International Conference on Vector and Parallel Processing*, volume 1573 of *Lecture Notes in Computer Science*, pages 43–57, 1999.

[4] Lefvert T. Ringhals i stability benchmark - final report. Technical Report NEA/NSC/DOC(96)22, OECD Nuclear Energy Agency, Paris, France, 1996.

[5] Satish Balay, William D. Gropp, Lois C. McInnes, and Barry F. Smith. Petsc home page. http://www.mcs.anl.gov/petsc, 2001.

[6] Satish Balay, William D. Gropp, Lois C. McInnes, and Barry F. Smith. Petsc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 1997.

[7] Satish Balay, William D. Gropp, Lois C. McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A.M. Bruaset, and H.P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Nirkhauser Press, 1997.

[8] Groupp W., Lusk E., and Skjellum A. *Using MPI: Portable Parallel Programming with Message Passing Interface*. MIT Press, 1994.

[9] Y. Saad. Sparskit: A basic toolkit for sparse matrix computations. Technical Report 90-20, Research Institute for Advanced Computer Science, NASA Ames Research Center, 1990.

[10] V. Kumar, A. Grama, A. Gupta, and G. Karypis. *Introduction to parallel computing:design and analysis of parallel algorithms*. The Benjamin/Cummings Publishing Company, Inc., Redwood City, CA, 1994.

[11] M. de J. Castel de Haro. *Métodos Iterativos Paralelos para la Resolución de Sistemas Lineales Hermíticos y Definidos Positivos*. PhD thesis, Universidad de Alicante, Valencia, España, 2000.