Algoritmo Tipo «Estrella» Para Resolver en Paralelo un Sistema de Ecuaciones Lineales Utilizando el Método de Householder

M. en C. Héctor Samuel García Salas
Profesor Investigador del CIDETEC-IPN
M. en C. Teodoro Alvarez Sánches
Profesor Investigador del CIC-IPN
[talvarez@pollux.cic.ipn.mx]

Este trabajo presenta una solución para resolver un sistema de ecuaciones lineales por medio del método de Householder, teniendo como base un algoritmo que utiliza en paralelo «n» procesadores, y como medio de comunicación entre los procesadores una interfaz MPI. El Programa que se desarrollo fue escrito para el compilador mpcc (Compilador de C para paso de mensajes de la interfaz MPI), para la máquina IBM SP2 en el Centro de Investigación en Computación (CIC) del IPN.

Antecedentes

"La forma de paralelizar la solución de un problema cae dentro de la categoría de lo artesanal y depende mucho del autor y de su experiencia". Esta cita muestra lo difícil que es encontrar el algoritmo ¢óptimo para llevar a cabo la paralelización de la ejecución de un programa, ya que no existe una metodología establecida y se depende casi absolutamente del punto de vista del autor del algoritmo. Algunos algoritmos son mas eficientes que otros; inclusive existen compiladores que permiten paralelizar automáticamente ciertas partes de programas secuenciales. Sin embargo, esto no los hace tan eficientes como podría esperarse.

Existen, por lo tanto, diferentes acercamientos y diferentes soluciones a un mismo problema. Brinch Hansen [Han95] trata la reducción de Householder utilizando un algoritmo de entubamiento (Pipelinening) desarrollado para Super Pascal [HAN95 pag.68]. Un enfoque diferente es la utilización de todos los procesadores para el cálculo de las n columnas de la matriz de Householder en cada etapa. Esto puede hacer mas eficiente el programa, y el tráfico en la red no se modifica en forma importante.

Planteamiento del Problema

Dado un sistema de ecuaciones lineales, uno de los métodos más utilizados para su solución es la eliminación gaussiana. Este método utiliza la matriz asociada de coeficientes y la matriz de resultados, para generar una matriz aumentada. Esta última matriz se lleva a la forma de una matriz triangular, y así se puede resolver el sistema por medio de:

$$x_i = \tfrac{1}{a_{ij}} [b_i - \sum_{j_{i+1}}^{n} a_{ij}\, x_j \;]$$

Sin embargo, el método de eliminación de Gauss presenta un problema de inestabilidad en el pivote; es decir, que cuando se tienen números muy pequeños o muy grandes se obtienen soluciones erróneas. Ahora bien, el método de Householder utiliza el mismo algoritmo que la eliminación Gaussiana, pero a nivel del pivote se hace una normalización con respecto a la columna sobre la cual se está trabajando.

Este método utiliza una argucia matemática, que consiste en la reflexión de un vector sobre un plano particular, de tal forma que el plano refleja la primera columna sobre el primer eje del sistema coordenado, para producir una nueva columna que tiene ceros como elementos después del primero. La **figura 1** muestra el esquema utilizado.

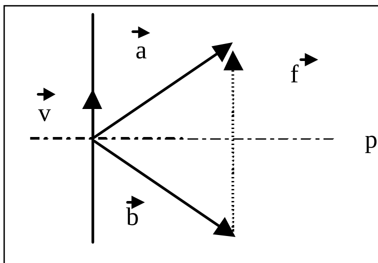


Figura 1

Dado un vector $\vec{a}$, existe un vector de reflexión $\vec{b}$ tal que:

$$\vec{a} - \vec{b} = f\,\vec{v} \text{ y } \|a\| = \|b\| \quad \ldots\ldots\ (1)$$

donde la normal de v esta dada por:

$$\|v\|=1 \text{ y } \|v\|=\sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

expresando al vector en forma matricial, se tiene:

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \quad y \quad v^t = \left[v_1 + v_2 + \cdots + v_n\right]$$

donde v^t es la matriz transpuesta del vector «v».

Ahora bien, el producto escalar entre la matriz del vector y su transpuesta es:

$$\begin{aligned} v^t v &= v_1^2 + v_2^2 + \cdots + v_n^2 \\ &= \| v \|^2 \end{aligned}$$

Para encontrar el vector equivalente, el cual contiene un solo elemento, se procede como sigue: de la expresión (1) :

$$b = a - f v \qquad ..... (2)$$

por la izquierda en la expresión 1:

$$\| a \|^2 = \| b \|^2 = b^t b$$

en forma matricial:

$$\begin{aligned} \| a \|^2 &= (a - f v)^t (a - f v) \\ &= (a^t - f v^t) (a - f v) \\ &= a^t a + f^2 v^t v - f a v^t - f a^t v \\ &= \| a \|^2 + f^2 \| v \|^2 - 2 f v^t a \end{aligned}$$

Simplificando ambos lados:

$$\| a \|^2 - \| a \|^2 = - 2 f v^t a + f^2 = 0$$

$$\therefore v^t a = f \qquad ...... (3)$$

Por la derecha en la expresión 1:

$$\| b \|^2 = \| a \|^2 = a^t a$$

En forma matricial:

$$\begin{aligned} \| b \|^2 &= (b + f v)^t (b + f v) \\ &= (b^t + f v^t) (b + f v) \\ &= b^t b + f^2 v^t v + f b v^t + f b^t v \\ &= \| b \|^2 + f^2 \| v \|^2 + 2 f v^t b \end{aligned}$$

Reduciendo ambos lados:

$$\| b \|^2 - \| b \|^2 = 2 f v^t b + f^2 = 0$$

$$\therefore v^t b = - f \qquad ...... (4)$$

Por otro lado, sustituyendo (3) en (2) se tiene que:

$$\begin{aligned} b &= a - f v = a - v f = a - v (2 v^t a) \\ &= a - 2 v v^t a = (I - 2 v v^t) a \\ &= H a \qquad ...... (5) \end{aligned}$$

Donde H es la matriz de Householder. Ahora bien, como

$$\| b \| = \| a \|$$

sustituyendo en (5)

$$\| H a \| = \| a \| \Rightarrow \quad a = H a$$

$$Ha = H(Ha) = a \Rightarrow H H = I \;\; y \;\; H = H^{-1}$$

Por otro lado :

$$| a_{ij} | \leq \| a_i \| \quad \forall \quad i, j = 1, ..., n$$

Ahora bien, regresando al sistema de ecuaciones lineales, y utilizando la propiedad de conservación del producto por la izquierda de una matriz se tiene el sistema

$$A x = y,$$

Donde "A" es la matriz de coeficientes, "x" es la matriz de incógnitas y "y" es la matriz de valores independientes.

Si "H" es la matriz de Householder entonces

$$H A x = H y$$

Tomando el primer vector de la matriz de coeficientes

$$a1 = [a11, a21, a31, ..., an1]^t \quad ;$$

entonces, aplicando la matriz de Householder para obtener el vector con un solo elemento :

$$H a_1 = \left[d_{11}, 0, \cdots, 0\right]^t$$

donde

$$d_{11} = \pm \| a_1 \|$$

de la expresión (1) se tiene:

$$f_1 v = a_1 - b_1$$

utilizando la expresión (4)

$$f_1 v = a_1 - Ha_1$$

$$= \begin{bmatrix} a_{11} - d_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} w_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix}$$

utilizando la expresión (4)

$$f_1 = - 2 v^t b_1 = - 2 v^t H a_1$$

multiplicando por f_1

$$f = f_1 (- 2 v^t H a_1) = - 2 f_1 v^t H a_1$$

$$= - 2 \begin{bmatrix} w_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} \left[d_{11}, 0, \cdots, 0\right]^t$$

$$= - 2 w_{11} d_{11}$$

$$f_1 = \sqrt{- 2 w_{11} d_{11}}$$

$$f_1 v = f_1 v \;\Rightarrow\; v = f_1 v / f_1$$

$$\therefore v = \left[w_{11}, a_{21}, \cdots, a_{n1}\right]^t / f_1 \qquad ...... (6)$$

De esta manera se ha encontrado el vector "v" que permite encontrar las columnas correspondientes a los otros vectores de la matriz de coeficientes, formadas por los vectores de «reflexión» correspondientes.

Como el vector "v" es invariable, entonces se pueden establecer las siguientes expresiones:

$$H\, a_i = a_i - f_i\, v \quad y \quad f_i = 2\, v^t\, a_i \qquad \text{...... (7)}$$

Estas expresiones, aplicadas a todos los vectores columna de la matriz original, permitirán encontrar la matriz equivalente, la cual tiene en su primera columna un solo elemento: el primero.

Con la matriz equivalente se puede continuar el procedimiento con la matriz disminuida de la primera columna y de la primera fila, hasta obtener una matriz disminuida equivalente con las características arriba mencionadas. El procedimiento se repite hasta obtener una matriz triangular que permita resolver el sistema de ecuaciones lineales.

Método y algoritmo de solución computacional

El método que se presenta aquí tiene como idea principal el utilizar varios procesadores que trabajando en paralelo permitan obtener de un sistema de ecuaciones lineales, la matriz triangular utilizando el método de Householder. Ahora bien, como se vió en la parte teórica, la obtención de la matriz triangular pasa por la obtención de varias matrices equivalentes, en las que, según sea el nivel de cálculo, se obtiene una primera columna relativa a la matriz disminuida con la que se está trabajando. Para el cálculo de cada matriz equivalente, este método pone a trabajar a todos los procesadores (la forma de trabajo se explica posteriormente).

Para que la idea de este método funcione, es necesario establecer el número de procesadores activos, utilizar uno de ellos como un procesador maestro, y a los otros como procesadores esclavos. La asignación de las tareas deberá efectuarla el proceso maestro; este determina el número de columnas que cada proceso esclavo debe calcular para cada nivel, siendo el proceso maestro el que distribuye, recolecta y ordena la información.

El algoritmo de paralelización que se presenta aquí utiliza un arreglo de cuatro procesadores. Dadas las características de MPI, la configuración para la comunicación de estos procesadores es en estrella, siendo el proceso que corre en el procesador cero el proceso maestro y los procesos en los otros tres procesadores los procesos esclavos. En la **figura 2**, se muestra el esquema de comunicación entre los procesos.

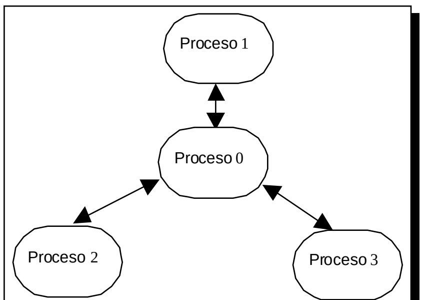


Figura 2

Algoritmo.

Para que el cálculo de la matriz equivalente en turno sea llevado a cabo en forma paralela por todos los procesadores asignados, los pasos a seguir son los que se describen a continuación:

- Se obtiene el número de procesos de acuerdo con el número de procesadores asignados.
- Se utiliza una matriz principal de tamaño nxn, que se maneja por el proceso maestro, y una matriz temporal del mismo tamaño en cada proceso.
- Se utiliza una matriz de tamaño 1xn para manejar los valores independientes del sistema.
- Se establece un canal de comunicación entre cada proceso esclavo y el proceso maestro.
- En un inicio, se llena la matriz principal con los valores de la matriz asociada a los coeficientes del sistema de ecuaciones a resolver, y la matriz de valores independientes.
- Se calcula el número de columnas a procesar por cada uno de los procesos esclavos.
- Utilizando las ecuaciones (3), se calculan los parámetros utilizados en la obtención de los vectores columna normalizados, siendo la primer columna la columna «pivote» (la cual tiene solo un elemento) y de la cual se obtienen tales parámetros.
- Se distribuyen los parámetros de cálculo y las columnas correspondientes a los procesadores respectivos.
- Se obtiene la matriz equivalente para los valores independientes.
- Se recolecta la información obtenida por cada procesador, se forma la matriz equivalente y se almacena.
- Se obtiene la siguiente matriz disminuida, y se repiten los pasos anteriores de cálculo, hasta que la matriz disminuida tenga un tamaño de 2 x 2.
- Una vez obtenida la matriz de Householder, se obtienen los valores de las variables independientes.

De acuerdo con el algoritmo anterior, las tareas que cada procesador debe efectuar son las siguientes:

El proceso maestro efectúa las siguientes tareas:

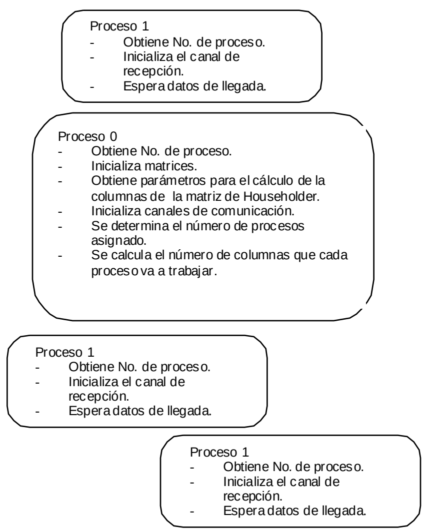
1. Llena la matriz inicial y la matriz temporal
2. Calcula el número de veces que se va a comunicar con los procesadores esclavos.
3. Se encarga de calcular los parámetros necesarios (el vector «v», ecuación [7]) para obtener las columnas que forman la matriz de Householder.
4. Calcula el número de columnas que cada proceso esclavo debe procesar.
5. Establece los canales de comunicación con los procesos esclavos.
6. Envía a cada uno de los procesos esclavos los par□metros calculados y las columnas correspondientes.
7. Con los par□metros correspondientes a la matriz en turno, procesa la matriz de valores independientes.
8. Colecta las columnas calculadas por los procesos esclavos.
9. Ordena y almacena la matriz de Householder.
10. Llama a la subrutina que proporciona la matriz disminuida siguiente a calcular.
11. Repite los pasos 2-7 hasta que la matriz disminuida es de tamaño 2 x 2.
12. Resuelve para la matriz de incógnitas.

Los procesos esclavos efectúan las siguientes tareas.

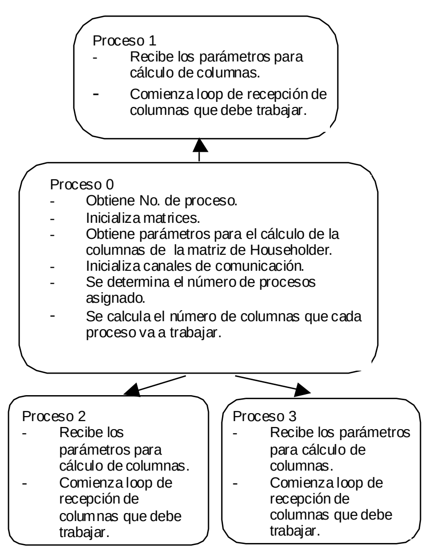
1. Calculan cuantas veces deben recibir datos.
2. Reciben los datos necesarios para llevar a cabo el proceso de las columnas.
3. Reciben las columnas que deben procesar.
4. Con los par□metros recibidos obtienen la columna respectiva de la matriz de Householder para cada columna recibida del proceso maestro.
5. Una vez terminados los cálculos sobre cada columna, esta es enviada de regreso al proceso maestro.
6. Repiten los pasos 2-4 hasta terminar.

Estados por los que pasa el programa que ejecuta el algoritmo anterior.

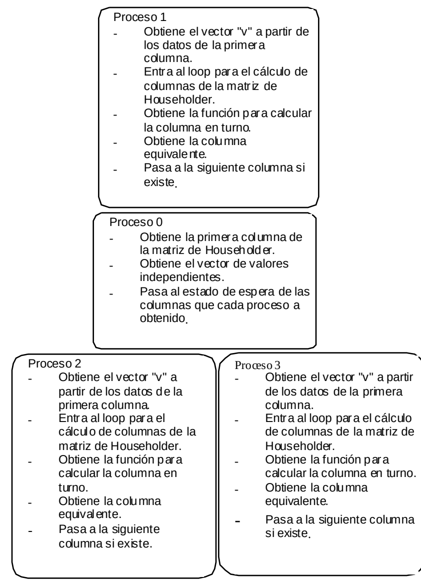
Primer etapa: Inicio



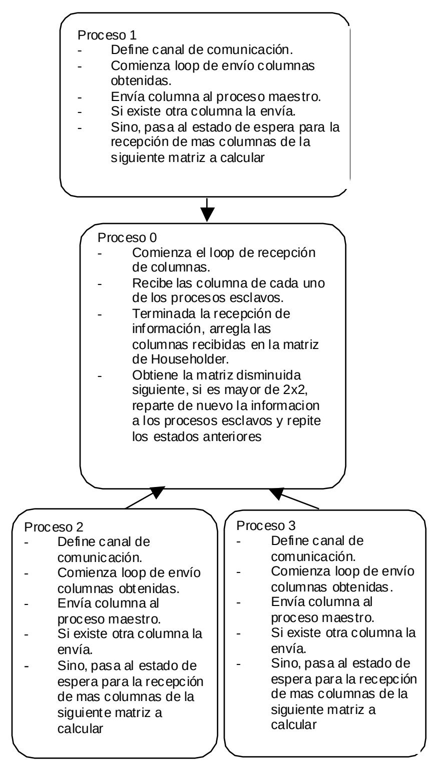
Siguiente etapa: envío de parámetros



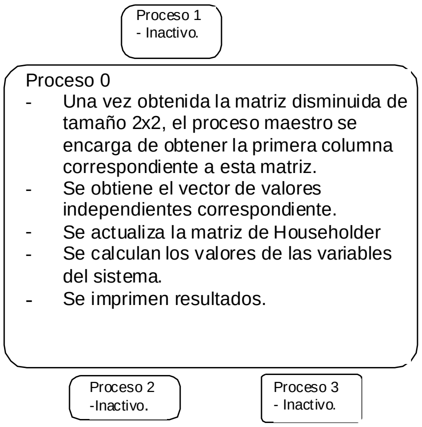
Siguiente etapa: Cálculo de columnas de la matriz de Householder



Siguiente etapa: envío de las columnas obtenidas por los procesos esclavos y recepción de esta información por el proceso maestro.



Etapa final: obtención de los valores de variables del sistema.



Parámetros de ejecución.

La IBM SP2 del CIC-IPN es una máquina con 8 procesadores, la cual necesita de un archivo de configuración donde se seleccionan los nodos en que se instalarán los procesos y los canales de comunicación que va a utilizar. Para esto es necesario un archivo que se asigna a la variable MP_HOSTFILE, mediante la instrucción:

```
%setenv MP_HOSTFILE SWITCH
```

donde SWITCH es el archivo que define las direcciones de los nodos que estarán activos.

La IBM SP2 tiene tres maneras de comunicar entre los ocho nodos: a través de una red externa (direcciones INTERNET), a través de una red interna (direcciones IP locales) o a través de una red de interconexión tipo mariposa HPS (SWITCH)

Las direcciones IP o dominios de tales redes se muestran en al final del artículo.

El número de procesadores que se desea estén activos cuando la aplicación se ejecute también se define en una variable de ambiente, que se inicializa de la siguiente manera:

```
%setenv MP_PROCS 4
```

donde 4 es el número de procesadores que se designarán a la aplicación.

La compilación de los programas se llevó a cabo con el compilador C para paso de mensajes MPI, con que cuenta la IBM SP2. El comando utilizado es:

```
%mpcc name.c /usr/lib/libm.a
```

y para la medición del tiempo de ejecución se utilizo el comando time:

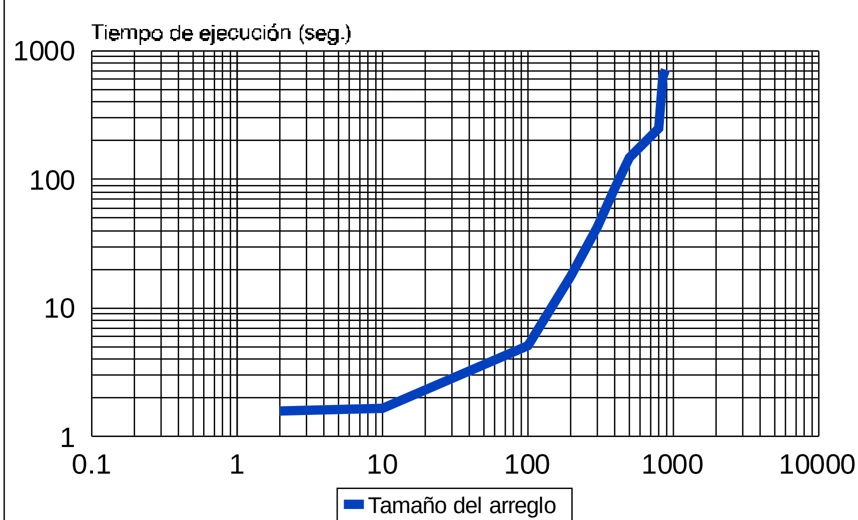
```
%/usr/bin/time -p ./a.out
```

Estos comandos evidentemente son diferentes para cada máquina paralela y para compilador que este montado.

Resultados

El programa se ejecutó para diferentes tamaños de matrices desde (2x2 hasta 400x400). Los tiempos de ejecución fueron en promedio los siguientes:

Tamaño	Tiempo real de ejecución (seg.)
2x2	1.60
10x10	1.65
100x100	5.10
200x200	17.50
300x300	42.25
400x400	85.30
500x500	148.77



Gráfica logarítmica del tiempo de ejecución (eje y) contra el tamaño del arreglo (eje x)

Conclusiones

Como puede observarse en la gráfica logarítmica, el tiempo de ejecución aumenta en forma drástica (pendiente de la recta de aproximación muy grande), a partir de un cierto tamaño de matriz de coeficientes (a partir de 100). Esto se debe a la cantidad de mensajes que circulan en la red de intercomunicacón

Al llegar a un tamaño cercano a los 900x900 la memoria se satura y las comunicaciones entre procesos causan errores.

Para corroborar los tiempos de ejecución con otros algortimos, se pretende implementarlos y llevar a cabo las corridas correspondientes. Sin embargo, el desarrollo del alorit-mo mostrado en este artículo muestra que, efectivamente, existen formas diferentes de enfrentar un problema, además de popder desarrollar una solución; y por otro lado, tampoco existe una mecánica para el desarro-llo de tal algoritmo.

Apéndice A

Direcciones de la red externa:

148.204.211.237
148.204.211.238
148.204.211.239
148.204.211.240
148.204.211.241
148.204.211.242
148.204.211.243
148.204.211.244

Direcciones de la red interna:

125.5.5.1
125.5.5.2
125.5.5.3
125.5.5.4
125.5.5.5
125.5.5.6
125.5.5.7
125.5.5.8

Direcciones de la red interna:

125.6.5.1
125.6.5.2
125.6.5.3
125.6.5.4
125.6.5.5
125.6.5.6
125.6.5.7
125.6.5.8

Bibliografía.

[1] [HAN95] Per Brinch Hansen. *"Studies in Computational Paradigms"*, 1995, Prentice Hall USA.

[2] [GRO96] William Gropp, E. Lusk, A. Skjellum. *"Using MPI"*. 1997, The MIT Press, England.

[3] [SAN97] Alfonso Sánchez Sandoval. *"Curso Básico de MPI"*. 1997, Proyecto final de Curso, CIC-IPN.