

RESOLUCION DE ECUACIONES EN DERIVADAS PARCIALES MEDIANTE ALGORITMOS PARALELOS

Crespo Fidalgo, J.L. Mora Monte, Eduardo Departamento de Matemática Aplicada y Ciencias de la Computación Universidad de Cantabria Peire Arroba, Juan Departamento de Ingeniería Eléctrica, Electrónica y de Control - UNED.

Resumen

En este artículo se propone una nueva metodología de resolución de ecuaciones en derivadas parciales basada en lo que modernamente se denominan procesos paralelos y distribuidos (PDP), también llamados sistemas complejos y redes neuronales.

Partiendo de una representación en diferencias finitas es posible llegar a un esquema de red en el que las conexiones se programan de antemano. El enfoque es idéntico con cualquier orden que se elija para la discretización por diferencias finitas, lo que da pie a un grado de aproximación variable. La resolución de este esquema en la red resulta equivalente a las técnicas de relajación para sistemas de ecuaciones lineales. Mediante ejemplos simples se ha comprobado la convergencia. Este enfoque permite aprovechar los recursos computacionales típicos de PDP para esta clase de problemas.

Palabras clave: Redes neuronales artificiales, procesamiento paralelo y distribuido, ecuaciones en derivadas parciales, sistemas conexionistas, multiproceso.

1. Introducción

En el Departamento de Matemática Aplicada y Ciencias de la Computación de la Universidad de Cantabria se están estudiando actualmente las posibilidades de los llamados algoritmos de proceso paralelo y distribuido. Concretamente, debido a un mayor conocimiento del área, se ha planteado la posibilidad de aplicarlos a la simulación de sistemas. Aunque este problema puede tratarse en forma completamente estadística (estilo modelos Box-Jenkins) es razonable pensar que un conocimiento añadido pudiera incrementar la bondad de los resultados. Lo que tienen en común los sistemas que son objeto de estudio en la ingeniería es que responden, en su mayoría, a ecuaciones en derivadas parciales, en el caso más general. En este artículo se plantea el primer escalón del estudio de estas aplicaciones, a saber: fijando el tipo de ecuación y el dominio, obtener un sistema que sea capaz de obtener una solución de las ecuaciones de campo en función de las condiciones de contorno.

No tiene sentido en este caso el aprendizaje por ejemplos, tal y como se ha utilizado con frecuencia con estos algoritmos, porque obligaría a resolver muchas veces el problema, con lo cual surge la paradoja de que para resolver el problema, este, se tiene que resolver previamente. En la sección 2 se plantea el origen del método, en la 3 la metodología, en las 4, 5 y 6 se aplica a distintos tipos de ecuaciones y en la 7 se extraen las conclusiones finales así como la línea a seguir en el futuro.

2. Origen del método

Clásicamente los problemas de ecuaciones en derivadas parciales se vienen resolviendo mediante las siguientes técnicas: diferencias finitas, elementos finitos, Montecarlo, Características, etc. Los problemas que se le plantean al ingeniero cuando tiene que resolver un problema son fundamentalmente: precisión requerida, coste computacional y estabilidad. La precisión viene condicionada por el orden del método, con el añadido de que un mayor orden va acompañado de mayor número de operaciones, lo que conduce a un error numérico mayor. Otra opción es mantener el orden pero reducir el tamaño de malla; esto mantiene bajo el error de truncamiento, pero incrementa el coste computacional.

La elección de un método u otro está condicionada por el tipo de ecuación a resolver y la geometría del dominio. Los programas más avanzados suelen emplear el método de elementos finitos, con generación automática de mallado. Esto reduce el esfuerzo del usuario, pero no afecta al coste computacional. La aplicación de las nuevas generaciones de ordenadores a estos problemas se ha realizado planteando métodos eficaces de resolución de sistemas de ecuaciones lineales mediante procesamiento paralelo o vectorial, teniendo en cuenta que casi siempre el problema numérico final es ese. Obsérvese que, por motivos históricos, se ha partido del método y se ha adaptado a la estructura computacional. Se considera ahora el camino inverso. Se parte de la estructura de resolución en paralelo, que se considera una de las líneas más prometedoras. El sistema más elemental de procesamiento en paralelo es el llamado proceso paralelo y distribuido (también llamados sistemas conectivistas, redes neuronales artificiales, etc.) Consisten en estructuras formadas por elementos interconectados; cada uno tiene varias entradas y una salida. Todo el proceso es local, es decir, un elemento no debe requerir de datos de elementos no conectados a l. Estos sistemas surgieron tratando de replicar la estructura de las redes de neuronas, si bien de forma muy simplificada. Incluyen una familia muy amplia de algoritmos que se distinguen

unos de otros en la forma de procesar cada elemento y en la forma de conectarse unos con otros. Normalmente los elementos de proceso se constituyen en grupos homogéneos, que pueden distinguirse por uno o ambos aspectos comentados. Cada grupo puede tener el significado de mecanismos de entrada o de salida, estructuras de proceso o de memoria, etc.

En resumen las características deseadas del algoritmo son: - Se ejecuta en un número elevado de procesadores (idealmente). - Los cálculos en cada procesador se realizan sobre variables locales a l. - La representación final es obra del conjunto de procesadores, por lo que sus "piezas" están distribuidas entre ellos. Con base a los estudios de Hopfield, [5] éstos modelos han comenzado a estudiarse desde el punto de vista de la mecánica estadística [2]. Así, se define una función de energía, que depende de la activación o valor de salida de cada elemento (llamado en este contexto el estado), otra de energía libre, entropía. La analogía parte de definir el estado del sistema como el conjunto de los estados de los elementos individuales. La temperatura es en este caso un parámetro que regula el ajuste de cada procesador elemental. Tenemos pues un método que está basado en la idea de procesamiento en paralelo; por tanto, y siempre que se respete su estructura, no tiene que preocuparse por cómo debe adaptarse a este tipo de arquitectura. Por otra parte el hecho de que pueda darse una interpretación mecánica implica que sus valores pueden ser representativos de variables físicas, del tipo de las que intervienen en las ecuaciones que queremos resolver.

3. Metodología

La idea de partida es que cada elemento de la red obtenga el valor de la función desconocida en un punto, y que la estructura de la función de activación y las conexiones sea tal que se verifique la ecuación diferencial que intentamos resolver. A continuación se desarrolla esta idea.

Se plantea una ecuación:

$$Df=0$$

siendo D un operador diferencial genérico.

Si se toma una aproximación del operador diferencial, se tendrá una ecuación en diferencias que relaciona el valor de f en distintos valores de sus variables:

$$L(f(x_1), f(x_2), \dots)=0$$

donde L es un operador que aproxima D y x_1, x_2, \dots son vectores de las variables en distintos puntos.

Si en este caso se supone que f es calculada por nuestros procesadores, se puede asignar a cada uno el valor de f en cada uno de los puntos ($f(x_i)$) (ver fig. 1) y obligar a que la estructura de la red, es decir las conexiones, sean tales que se verifique la ecuación anterior.

Concretamente, si L es una aproximación local y si f es despejable, la ecuación se podrá poner como:

$$f(\text{centro}) = G(f(x(1)), f(x(2)), \dots)$$

y esta misma ecuación ser válida en cada punto, tomando cada vez la aproximación local correspondiente.

Este planteamiento es totalmente análogo al desarrollado por el método de diferencias finitas, donde se plantean aproximaciones lineales de los operadores diferenciales.

Si el sistema de ecuaciones lineales resultante se resuelve siguiendo una vía interactiva (Gauss-Seidel, por ejemplo) se puede llegar a la misma expresión anterior.

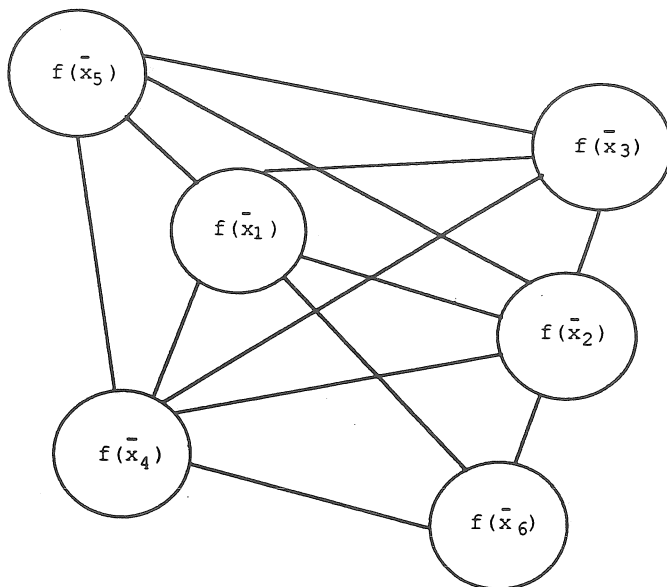


Fig. 1. Representación de un campo mediante una red de procesadores. Cada elemento calcula el valor de la función en un punto.

No obstante en la presentación que se ha hecho no se ha requerido que la aproximación fuera lineal, lo que le da un carácter más general. En concreto los requisitos para llegar a la última expresión son:

- Un operador en diferencias que aproxime el operador diferencial localmente.
 - El valor de la función en el centro de este entorno local debe poderse despejar explícitamente.
 - El sistema formado por todas estas ecuaciones ha de ser estable.
- El último requisito es meramente de resolución, para garantizar que a partir de una configuración cualquiera del sistema se puede llegar a la solución.

4. Aplicación a ecuaciones elípticas

Sea la ecuación $\Delta f = 0$ en un dominio 1-D.

Tomemos una aproximación por diferencias finitas de primer orden

$$f(x-\Delta) + f(x+\Delta) - 2f(x) = 0$$

Tomando una discretización con intervalo constante, se tiene para un nodo f_i

$$f_i = \frac{f_{i-1} + f_{i+1}}{2} = 0.5f_{i-1} + 0.5f_{i+1}$$

Sumando f_i a ambos lados y dividiendo por dos, para conseguir estabilidad se tiene:

$$f_i = 0.25f_{i-1} + 0.25f_{i+1} + 0.5f_i$$

donde el término f_i de la derecha corresponde al valor previo de la interacción.

Analíticamente se tiene

$$\frac{d^2f}{dx^2} = 0 \Rightarrow \frac{df}{dx} = K_1 \Rightarrow f = K_1x + K_2$$

Luego la solución de una recta que une las condiciones de contorno, por ejemplo el caso presentado en la fig. 2:

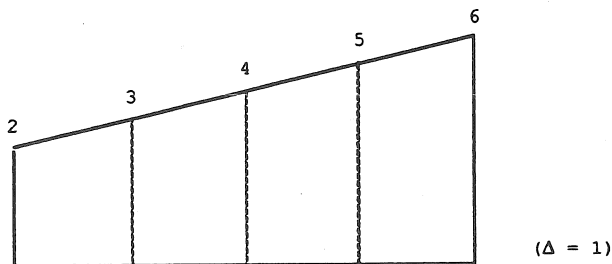


Fig. 2 Ejemplo elemental de ecuación diferencial de 2º orden resuelto analíticamente.

Los resultados de la red, programada según la expresión deducida anteriormente se presentan en la fig. 3:

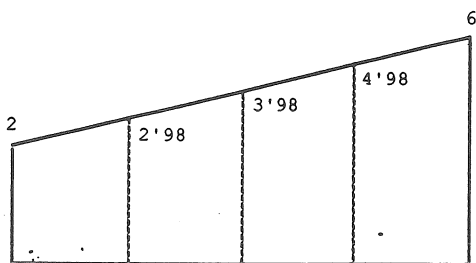


Fig. 3. Solución obtenida por la red para el mismo problema de la fig. 2.

Por lo que se deduce que la precisión está limitada sólo por el número de iteraciones.

Se tiene ahora un caso más complicado como el que se presenta con la ecuación

$$\frac{d}{dx} \left(K \frac{df}{dx} \right) + \frac{d}{dy} \left(K \frac{df}{dy} \right) = 0 \text{ con } K = K(\xi)$$

aproximándola como:

$$\frac{\frac{1}{2} \{K[T(x_1, y_0)] + K[T(x_0, y_0)]\} \frac{T(x_1, y_0) - T(x_0, y_0)}{x_1 - x_0} - \frac{1}{2} \{K[T(x_{-1}, y_0)] + K[T(x_0, y_0)]\}}{\frac{1}{2}(x_0 + x_1) - \frac{1}{2}(x_1 + x_0)}$$

$$+ \frac{\frac{T(x_0, y_0) - T(x_{-1}, y_0)}{x_0 - x_{-1}} + \frac{1}{2} \{K[T(x_0, y_1)] + K[T(x_0, y_0)]\} \frac{T(x_0, y_1) - T(x_0, y_0)}{y_1 - y_0}}{\frac{1}{2}(y_0 + y_1) - \frac{1}{2}(y_{-1} + y_0)} = 0$$

Se puede despejar, haciendo $\Delta x = \Delta y = cte$

$$T(x_0, y_0) = \frac{\{K[T(x_1, y_0)] + K[T(x_0, y_0)]\}T(x_1, y_0) + \{K[T(x_{-1}, y_0)] + K[T(x_0, y_0)]\}T(x_{-1}, y_0) + K[T(x_1, y_0)] + K[T(x_0, y_0)] + K[T(x_0, y_1)] + K[T(x_0, y_{-1})] + 4K[T(x_0, y_0)]}{\{K[T(x_0, y_1)] + K[T(x_0, y_0)]\}T(x_0, y_1) + \{K[T(x_0, y_{-1})] + K[T(x_0, y_0)]\}T(x_0, y_{-1})}$$

A esta ecuación se le aplicaría el método iterativo del caos anterior.
 En la fig. 4 pueden verse los resultados para un dominio y condiciones de contorno concretas

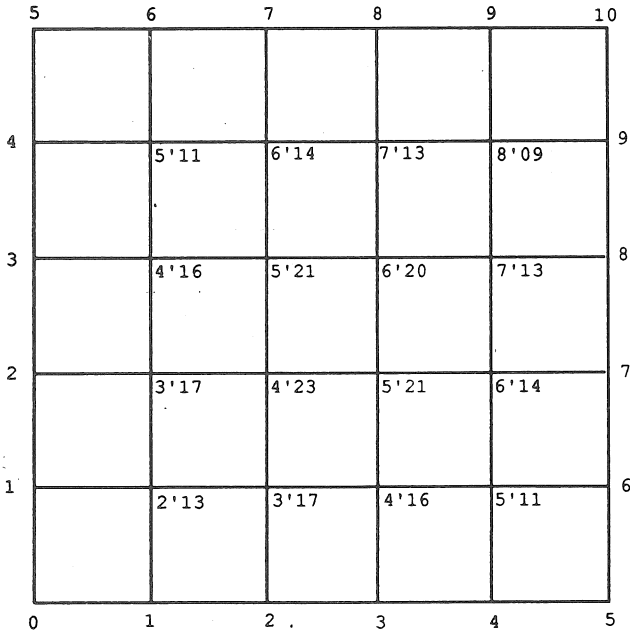


Fig. 4 Resolución de ecuación elíptica no lineal.

5. Aplicación a ecuación parabólica

Sea $\frac{d^2f}{dx^2} = K \frac{df}{dt}$ CON $K = \frac{K_0}{f}$

Utilizando la aproximación de Crank-Nicolson con proyección hacia adelante se llega a una expresión de la forma

$$f(x_0, t_1)^j = \beta \frac{\alpha \varepsilon \omega_j f(x_j, t_j) + K f(x_0, t_0)}{K + \omega_0 \alpha} + (1 - \beta) f(x_0, t_1)^{j-1}$$

donde el superíndice j se refiere a la iteración, β es el parámetro a elegir para las iteraciones, $\alpha = \frac{\Delta t}{\Delta x}$, ω_j son múltiplos enteros de $\frac{1}{\Delta x}$

Esto se puede solucionar para unas condiciones iniciales y en los extremos, por ejemplo, para:

$$\kappa = \frac{10}{f}, \quad \alpha = 0'01, \quad \Delta x = 0'02, \quad \beta = 0'5$$

Se obtendrían los resultados indicados en la fig. 5.

	10'1	12'3	14'5	16'7	18'9
10'1		12'3	14'25	15'88	16'47
9'99		11'97	13'63	14'77	15'47
9'98		11'64	13'05	14'09	14'60
9'95		11'38	12'60	13'47	13'93
9'93		11'17	12'22	12'98	13'37
9'91		10'99	11'91	12'57	12'92
9'90		10'84	11'64	12'22	12'53
9'89		10'71	11'42	11'93	12'20
9'88		10'60	11'23	11'68	11'92
9'87		10'51	11'06	11'46	11'68

Fig. 5. Resolución de una ecuación parabólica no lineal con condiciones simétricas.

En el sistema hay en cada momento una capa para el instante anterior y otra para el actual

6. Aplicación a ecuación hipérbola

Sea $\frac{d^2f}{dx^2} = \frac{d^2f}{dt^2}$

Suponiendo $\Delta x = \Delta t = \text{cte}$ se puede aproximar como

$$f(x, t_0) - f(x_0, t_0) - [f(x_0, t_0) - f(x_1, t_0)] = f(x_0, t_1) - f(x_0, t_0) - [f(x_0, t_0) - f(x_1, t_{-1})]$$


de donde se puede despejar $f(x_i, t_i)$ y resolver por relajación como en los casos anteriores.

La red tendrá 3 capas: una para t_{-1} , otra para t_0 y otra para t_1 .

Tal y como se ha planteado, la aproximación está en el límite de estabilidad.

Los resultados para un caso concreto pueden verse en la figura 6:

	0	0'2	0'3	0'1	0'2	0'4	0'5	0'3	0'5	0'2	
0'1		0'10	-0'00	0'4	0'3	0'3	0'2	0'7	0		0'3
0		-0'1	0'2	0'2	0'5	0'1	0'5	-0'1	0'5		0'2
0		0'1	0'1	0'3	0	0'7	-0'2	0'3	0'1		0'2
0		0'2	0'2	-0'1	0'5	-0'3	0'5	0	0		0'2
0		0'1	0	0'4	-0'39	0'29	-0'09	0'2	0'1		0'2
0		-0'2	0'3	-0'29	0'19	-0'19	-0'01	0'01	0'4		0'2
0		0'19	-0'49	0'09	-0'09	-0'11	-0'09	0'19	0'1		0'2
0		-0'29	-0'01	-0'28	-0'21	0'01	0'09	0'01	0		0'2
0		-0'21	-0'08	-0'31	-0'18	-0'01	0'11	-0'1	0'1		0'2
0		0'2	-0'51	0'02	-0'11	-0'08	-0'2	0'2	0'1		0'2
0		-0'3	0'3	-0'31	0'12	-0'31	0'01	0	0'3		0'2



t

Fig. 6 Resolución de ecuación hiperbólica sencilla.

El paso de tiempo utilizado es el máximo estable.

7. Conclusiones

Se ha presentado un enfoque que permite compaginar los algoritmos de proceso paralelo y distribuido con la resolución de ecuaciones en derivadas parciales en paralelo. Esta idea, inspirada en la analogía entre elementos de proceso y partículas, tal y como se estudian en Mecánica Estadística, puede presentar un amplio campo de aplicación. Los casos prácticos aquí planteados tienen como caso particular la resolución interactiva de aproximaciones en diferencias finitas, lo que se ha demostrado con varios ejemplos. Ha quedado patente la

forma de plantear la analogía mencionada, y se ha comprobado, a nivel de los ejemplos, la utilidad del método que permite sustituir la programación específica de ordenadores multiproceso por la de un sistema PPD. Con el desarrollo de los nuevos ordenadores basados en técnicas del PDP, se abrirían nuevos campos de aplicación y en especial en los problemas más complejos dentro del ámbito de la ingeniería.

Referencias

- [1] J.L. McClelland y D.E. Rumelhart, Explorations in Parallel Distributed Processing, MIT Press, 1989.
- [2] J. Hertz, A. Krogh y R. G. Palmer, Introduction to the theory of neural computation, Addison-Wesley, 1991.
- [3] M. Quinn. Designing Efficient Algorithms for Parallel Computers. McGraw-Hill, 1987.
- [4] J. M. Ortega, R.G. Voigt, Solution of Partial Differential Equations on Vector and Parallel Computers, SIAM, 1985.
- [5] Hopfield, J.J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proceedings of the National Academy of Sciences, USA 79, 2554-2558. Reprinted in Anderson and Rosenberg [1988].