Revista de Matemática: Teoría y Aplicaciones 2012 **19**(2) : 141–156
cimpa – ucr issn: 1409-2433

IMPLEMENTACIÓN DEL MÉTODO LDG PARA MALLAS NO ESTRUCTURADAS EN 3D

IMPLEMENTATION OF LDG METHOD FOR 3D UNSTRUCTURED MESHES

Filánder A. Sequeira Chavarría* Paul E. Castillo†

Received: 21 Feb 2011; Revised: 12 Jun 2012;
Accepted: 18 Jun 2012

---

*Escuela de Matemática, Universidad de Costa Rica, 2060 San José, Costa Rica. E-Mail: filander.sequeira@ucr.ac.cr

†Departamento de Ciencias Matemáticas, P.O. Box 9000, Universidad de Puerto Rico, 00681 Mayagüez Puerto Rico. E-Mail: paul.castillo@upr.edu

Resumen

En este artículo se describe una implementación del método "Local Discontinuous Galerkin" (LDG) aplicado a problemas elípticos en 3D. Se discute la implementación de los principales operadores. En particular el uso de aproximaciones de alto orden y de mallas no estructuradas. Estructuras de datos eficientes que permiten un rápido ensamblado del sistema lineal en su formulación mixta son descritas en detalle.

Palabras clave: Métodos de elemento finito discontinuos, aproximaciones de alto orden, mallas no estructuradas, programación orientada a objetos.

Abstract

This paper describes an implementation of the Local Discontinuous Galerkin method (LDG) applied to elliptic problems in 3D. The implementation of the major operators is discussed. In particular the use of higher-order approximations and unstructured meshes. Efficient data structures that allow fast assembly of the linear system in the mixed formulation are described in detail.

Keywords: Discontinuous finite element methods, high-order approximations, unstructured meshes, object-oriented programming.

Mathematics Subject Classification: 65K05, 65N30, 65N55.

1 Introducción

En las últimas décadas, se han propuesto métodos de alto orden que combinan las ideas de las técnicas de elemento finito y de volumen finito. Con el fin de preservar propiedades físicas del problema, se han diseño una serie de métodos llamados "Discontinuous Galerkin" (DG). En [2] se desarrolló un marco teórico abstracto bajo el cual se presentó un análisis de convergencia y de estabilidad de los métodos más importantes hasta esa fecha. En [5] se realizó una comparación de varios métodos DG para un problema modelo en 2D y se analizó el comportamiento del condicionamiento espectral de la matriz de rigidez. En este artículo trabajaremos con el método "Local Discontinuous Galerkin" (LDG), el cual fué propuesto en [9] para problemas transcientes de convección-difusión, y analizado en [7] para un problema modelo estacionario de difusión lineal. En [10] se presentó un análisis *hp* de convergencia para problemas lineales. El caso de difusión no lineal se discutió en [4].

Las implementaciones del método LDG conocidas hasta la fecha se restringen a mallas no estructuradas para dominios de $\mathbb{R}^2$. Para nuestro

conocimiento existe solamente una librería que implementa el método LDG en 3D: *Deal II* desarrollada por W. Bangerth, R. Hartmann y G. Kanschat en [3]. Sin embargo, esta librería utiliza únicamente mallas Cartesianas. En este trabajo presentamos una implementación del método LDG para dominios en 3D la cual utiliza mallas no estructuradas y polinomios de alto orden.

El paradigma de programación orientada a objetos provee de manera natural los mecanismos necesarios para una implementación desde un nivel abstracto. Como resultado se obtiene un código modular, fácil de mantener y que puede ser extendido por el usuario. El código se desarrolló utilizando el lenguaje de programación C++ y consiste de un conjunto de módulos independientes los cuales proveen una interfaz abstracta. De esta forma el código no está sujeto a clases concretas específicas como por ejemplo el uso de cuadraturas con una distribución de nodos simétrica o de bases de polinomios particulares. Las estructuras de datos relacionadas con el núcleo de los cálculos no se basan en características del paradigma orientado a objetos, por lo que estas pueden ser adaptadas a códigos de Fortran, por medio de la librería LAPACK [1].

El artículo está organizado como sigue: en la Sección 2 se presenta brevemente la formulación del método aplicado a un problema modelo; en la Sección 3 se discuten los operadores más importantes del método LDG. En la Sección 4 se presentan algunos experimentos numéricos con el propósito de validar el código y finalmente, se presentan algunas conclusiones en la Sección 5.

2 El método "Local Discontinuous Galerkin"

A continuación damos una breve descripción del método "Local Discontinuous Galerkin", el cual utilizaremos para aproximar la solución a problemas elípticos en 3D, con condiciones de frontera de Dirichlet, Neumann y Robin:

$$\begin{aligned} -\nabla \cdot \mathcal{K}\nabla u &= f, \quad \text{en } \Omega, \\ u &= g_{\mathcal{D}}, \text{ sobre } \partial\Omega_{\mathcal{D}}, \\ \mathcal{K}\nabla u \cdot \vec{\boldsymbol{n}} &= g_{\mathcal{N}}, \text{ sobre } \partial\Omega_{\mathcal{N}}, \\ au + \mathcal{K}\nabla u \cdot \vec{\boldsymbol{n}} &= g_{\mathcal{R}}, \text{ sobre } \partial\Omega_{\mathcal{R}}, \end{aligned}$$

donde $\mathcal{K}$ es un tensor a trozos simétrico positivo definido, Ω un dominio acotado de $\mathbb{R}^3$, y $\partial\Omega = \partial\Omega_{\mathcal{D}} \cup \partial\Omega_{\mathcal{N}} \cup \partial\Omega_{\mathcal{R}}$.

Introducimos una nueva variable $\boldsymbol{q} = \mathcal{K}\nabla u$, la cual en algunos casos, como en medios porosos, representa la velocidad del fluido, o mejor cono-

cida como la velocidad de Darcy, y la variable u representa la presión hidrostática. Al realizar el cambio de variable, obtenemos las ecuaciones de primero orden:

$$\begin{aligned} \mathcal{K}^{-1}\boldsymbol{q} &= \nabla u, \text{ en } \Omega, && (1)\\ -\nabla\cdot\boldsymbol{q} &= f, \quad \text{en } \Omega, && (2)\\ u &= g_D, \text{ sobre } \partial\Omega_{\mathcal{D}},\\ \boldsymbol{q}\cdot\vec{\boldsymbol{n}} &= g_{\mathcal{N}}, \text{ sobre } \partial\Omega_{\mathcal{N}},\\ au+\boldsymbol{q}\cdot\vec{\boldsymbol{n}} &= g_{\mathcal{R}}, \text{ sobre } \partial\Omega_{\mathcal{R}}. \end{aligned}$$

Sea $\mathcal{T}_h$ una partición del dominio Ω, la cual consiste de un conjunto de tetrahedros. La formulación débil del problema se obtiene multiplicando la ecuación (1) y (2) por funciones de prueba $\boldsymbol{r}$ y v respectivamente, donde $\boldsymbol{r}$ es una función vectorial y v una función escalar. A diferencia del elemento finito clásico integramos por partes en cada elemento $T\in\mathcal{T}_h$, de donde se obtiene la siguiente formulación débil:

$$\begin{aligned} \int_T \mathcal{K}^{-1}\boldsymbol{q}\cdot\boldsymbol{r} &= \oint_{\partial T} u\boldsymbol{r}\cdot\vec{\boldsymbol{n}}_T - \int_T u\nabla\cdot\boldsymbol{r}, && (3)\\ \int_T \boldsymbol{q}\cdot\nabla v &= \oint_{\partial T} v\boldsymbol{q}\cdot\vec{\boldsymbol{n}}_T + \int_T f v, && (4) \end{aligned}$$

donde $\vec{\boldsymbol{n}}_T$ es un vector normal unitario exterior al elemento T. Esta formulación no se restringe a mallas conformes como las que se utilizan en elemento finito clásico. La solución exacta $(u,\boldsymbol{q})$ se aproxima por el par $(u_h,\boldsymbol{q}_h)$ que se encuentran en $\mathcal{V}_h\times\mathcal{M}_h$, donde

$$\begin{aligned} \mathcal{V}_h &= \left\{u\in L_2(\Omega)\ :\ u|_T\in\mathcal{P}_{k_T}(T),\forall\, T\in\mathcal{T}_h\right\},\\ \mathcal{M}_h &= \left\{\boldsymbol{q}\in\left(L_2(\Omega)\right)^3\ :\ \boldsymbol{q}|_T\in\mathcal{P}_{k_T}(T)^3,\forall\, T\in\mathcal{T}_h\right\}. \end{aligned}$$

El espacio $\mathcal{P}_{k_T}(T)$ es el conjunto de polinomio de grado k_T en la celda T, donde $k_T\le k$, para k fijo. Las definiciones de estos conjuntos permiten que los grados de los polinomios puedan ser diferentes en cada elemento $T\in\mathcal{T}_h$.

La solución discreta $(u_h,\boldsymbol{q}_h)$ es definida usando la formulación débil anterior para cada $T\in\mathcal{T}_h$ y $(v,\boldsymbol{r})\in\mathcal{V}_h(T)\times\mathcal{M}_h(T)$ de la siguiente manera:

$$\int_T \mathcal{K}^{-1}\boldsymbol{q}_h \cdot \boldsymbol{r} - \oint_{\partial T} \widehat{u_h}_{\{f,T\}} \boldsymbol{r} \cdot \vec{\boldsymbol{n}}_T + \int_T u_h \nabla \cdot \boldsymbol{r} = 0, \quad (5)$$

$$\int_T \boldsymbol{q}_h \cdot \nabla v - \oint_{\partial T} v \widehat{\boldsymbol{q}_h}_{\{f,T\}} \cdot \vec{\boldsymbol{n}}_T + \oint_{\partial T} v \alpha_f(u_h) \cdot \vec{\boldsymbol{n}}_T = \int_T f v, \quad (6)$$

donde $\widehat{u_h}_{\{f,T\}}$ y $\widehat{\boldsymbol{q}_h}_{\{f,T\}}$ son llamados *flujos numéricos*, los cuales aproximan las trazas de las funciones u y $\boldsymbol{q}$, respectivamente, en la cara f del elemento T. El término $\alpha_f(u)$ juega el papel de parámetro de estabilidad, el cual garantiza la existencia de la solución del problema discreto bajo ciertas condiciones.

Sea f una cara interior, la cual es compartida por los elementos T y K. Los flujos numéricos para el LDG vienen dados de la siguiente manera:

$$\begin{aligned} \widehat{u_h}_{\{f,T\}} &= \{u_h\} + \boldsymbol{\beta}_f \cdot [\![u_h]\!], \\ \widehat{\boldsymbol{q}_h}_{\{f,T\}} &= \{\boldsymbol{q}_h\} - \boldsymbol{\beta}_f \cdot [\![\boldsymbol{q}_h]\!], \\ \alpha_f(u_h) &= \eta_f [\![u_h]\!], \end{aligned}$$

donde $[\![v]\!]$ y $\{v\}$ denotan el *salto* y el *promedio* (respectivamente) de la función v, es decir:

$$[\![v]\!] = v|_K \vec{\boldsymbol{n}}_K + v|_T \vec{\boldsymbol{n}}_T \quad \text{y} \quad \{v\} = \frac{1}{2}(v|_K + v|_T),$$

los parámetros $\boldsymbol{\beta}_f$ y η_f dependen de la cara. La convergencia y la estabilidad del método LDG fueron probadas en [7] para $\eta_f > 0$. La definición de los flujos numéricos para una cara de frontera f, se muestra en la siguiente cuadro:

Flujo	Dirichlet	Neumann	Robin
$\widehat{u_h}_{\{f,T\}}$	$g_{\mathcal{D}}$	u_h	u_h
$\widehat{\boldsymbol{q}_h}_{\{f,T\}} \cdot \vec{\boldsymbol{n}}_T$	$\boldsymbol{q}_h \cdot \vec{\boldsymbol{n}}_T$	$g_{\mathcal{N}}$	$g_{\mathcal{R}} - a u_h$
$\alpha_f(u_h)$	$\eta_f (u_h - g_{\mathcal{D}}) \vec{\boldsymbol{n}}_T$	0	0

Tabla 1: Definición de los flujos numéricos en la frontera.

3 Sistema lineal en su forma mixta

El sistema lineal generado por el método LDG tiene la estructura de bloques:

$$\begin{pmatrix} D & B \\ -B^T & S \end{pmatrix} \begin{pmatrix} \boldsymbol{q}_h \\ u_h \end{pmatrix} = \begin{pmatrix} b_q \\ b_u \end{pmatrix}, \tag{7}$$

donde las matrices D, B, $-B^T$ y S son las versiones discretas de la difusión, gradiente, divergencia y estabilidad, respectivamente.

En la práctica se resuelve para la variable primaria u_h. Esto se puede hacer mediante eliminación Gaussiana por bloques, obteniendo el siguiente sistema mejor conocido como el complemento de Schur $\mathcal{A}$:

$$\mathcal{A}u_h = b_u + B^T D^{-1} b_q, \text{ donde } \mathcal{A} = S + B^T D^{-1} B.$$

En [7] se demostró que $\mathcal{A}$ es simétrica definida positiva y en [5] se demostró que su condicionamiento espectral $\kappa(\mathcal{A}) = \mathcal{O}(h^{-2})$ donde $h = \max\{\text{diam}(T) : T \in \mathcal{T}_h\}$. El comportamiento asintótico de $\kappa(\mathcal{A})$ es igual al del método de elemento finito clásico. Como se verá en la Sección 3.1 la matriz D puede ser invertida explícitamente, lo cual resulta muy conveniente para la obtención del complemento de Schur de manera directa.

En esta sección describimos en detalle el cálculo de las matrices D, B y S tanto para mallas estructuradas como no estructuradas. Por simplicidad, se asume que el grado de aproximación es el mismo en todas las celdas. Se consideran únicamente aquellas mallas cuyas celdas son equivalentes mediante una transformación lineal o afín a una celda fija llamada celda de referencia. En otras palabras, se denota por $\hat{T}$ la celda de referencia entonces para cada celda $T_k \in \mathcal{T}_h$ existe un mapeo afín $\phi_k(\hat{x}) = \mathcal{J}_k\hat{x} + b$ tal que $\phi_k(\hat{T}) = T_k$, donde el Jacobiano $\mathcal{J}_k$ satisface $|\mathcal{J}_k| = \det\mathcal{J}_k > 0$.

Para cada $T_k \in \mathcal{T}_h$ denotamos por $\Phi = \{\varphi_i^k\}$ una base para el espacio local de polinomios $\mathcal{V}_h(T_k)$ (variable primaria o potencial) y por $\boldsymbol{\Psi} = \Phi \times \Phi \times \Phi$ una base para el espacio local de polinomios $\mathcal{M}_h(T_k)$ (variable secundaria o gradiente). Obsérvese que estas bases pueden ser construidas a partir de bases en el elemento de referencia ($\varphi_i^k = \hat{\varphi}_i \circ \phi_k^{-1}$). Finalmente, se denota por N el número total de celdas en $\mathcal{T}_h$.

3.1 Operador de difusión

Sea $\mathcal{K}_k$ el tensor de difusión de la celda T_k, el cual se asume constante en cada celda. Entonces la representación matricial del operador discreto de difusión relativo a la base $\boldsymbol{\Psi}$ es una matriz diagonal de $N \times N$ bloques. Además, como $\boldsymbol{\Psi}$ es un producto de bases escalares, cada bloque de la diagonal puede ser factorizado mediante el producto Kronecker de dos matrices como se muestra a continuación

$$D_{kk} = \left[\int_{T_k} \boldsymbol{\psi}_i^k \cdot \mathcal{K}_k^{-1} \boldsymbol{\psi}_j^k\right] = |\mathcal{J}_k| \mathcal{K}_k^{-1} \otimes M,$$

donde $M = \left[\int_{\hat{T}} \hat{\varphi}_i \hat{\varphi}_j\right]$ es la matriz de masa asociada a la celda de referencia $\hat{T}$. Observe que siendo D una matriz diagonal por bloques esta puede ser invertida fácilmente. Además la inversa de cada bloque de la diagonal es también un producto Kronecker:

$$D_{kk}^{-1} = \left[\int_{T_k} \boldsymbol{\psi}_i^k \cdot \mathcal{K}_k^{-1} \boldsymbol{\psi}_j^k\right]^{-1} = |\mathcal{J}_k|^{-1} \mathcal{K}_k \otimes M^{-1}.$$

Nótese que las dimensiones globales de las matrices D y D^{-1} son

$$N\dim\, \mathcal{M}_h(\hat{T}) \times N\dim\, \mathcal{M}_h(\hat{T}) \;=\; 3N\dim\, \mathcal{V}_h(\hat{T}) \times 3N\dim\, \mathcal{V}_h(\hat{T}).$$

Gracias al producto Kronecker y a que las celdas son linealmente equivalentes no es necesario almacenar explícitamente ninguna de estas matrices. Todo se reduce al cálculo de la matriz M^{-1}.

3.2 Operador gradiente

La representación discreta del gradiente está dada por la matriz B, la cual tiene una estructura esparcida de $N \times N$ bloques rectangulares cuyas dimensiones son $\dim\, \mathcal{M}_h(T_k) \times \dim\, \mathcal{V}_h(T_k)$. El operador gradiente asociado a la celda T_k se obtiene a partir de la siguiente expresión:

$$\int_{T_k} u_h \nabla \cdot \boldsymbol{\psi}_m^k - \oint_{\partial T_k} \widehat{u_{h\{f,T_k\}}} \boldsymbol{\psi}_m^k \cdot \vec{\boldsymbol{n}}_k. \tag{8}$$

Como todas las celdas son linealmente equivalentes a la celda de referencia $\hat{T}$, la contribución al bloque B_{kk} por el término que involucra la integral de volumen de la ecuación (8) puede se calculado como sigue:

$$\begin{aligned}
&\left[\int_{T_k} \varphi_n^k \nabla \cdot \boldsymbol{\psi}_m^k\right] = \\
&\quad = \quad |\mathcal{J}_k| \left(\begin{bmatrix} \partial_x \hat{x} \\ \partial_y \hat{x} \\ \partial_z \hat{x} \end{bmatrix} \otimes DX + \begin{bmatrix} \partial_x \hat{y} \\ \partial_y \hat{y} \\ \partial_z \hat{y} \end{bmatrix} \otimes DY + \begin{bmatrix} \partial_x \hat{z} \\ \partial_y \hat{z} \\ \partial_z \hat{z} \end{bmatrix} \otimes DZ \right),
\end{aligned}$$

donde

$$DX = \left[\int_{\hat{T}} \hat{\varphi}_n \partial_{\hat{x}} \hat{\varphi}_m\right], \;\; DY = \left[\int_{\hat{T}} \hat{\varphi}_n \partial_{\hat{y}} \hat{\varphi}_m\right] \quad \text{y} \quad DZ = \left[\int_{\hat{T}} \hat{\varphi}_n \partial_{\hat{z}} \hat{\varphi}_m\right].$$

Las matrices DX, DY y DZ se calculan sólamente para la celda de referencia. Los demás coeficientes pueden ser obtenidos de la información geométrica de la celda T_k,

$$\mathcal{J}_k^{-1} = \begin{bmatrix} \partial_x\hat{x} & \partial_y\hat{x} & \partial_z\hat{x} \\ \partial_x\hat{y} & \partial_y\hat{y} & \partial_z\hat{y} \\ \partial_x\hat{z} & \partial_y\hat{z} & \partial_z\hat{z} \end{bmatrix}.$$

Ahora, para la integral de superficie de la ecuación (8), consideramos f una cara interior, compartida por las celdas T_k y T_i. Sean u_h^k y u_h^i las aproximaciones de la solución en las celdas T_k y T_i respectivamente. El flujo numérico $\widehat{u_h}_{\{f,T_k\}}$ se define como una combinación lineal de estas cantidades de la siguiente manera:

$$\widehat{u_h}_{\{f,T_k\}} \;=\; \alpha_i u_h^k + \zeta_i u_h^i,$$

donde $(\alpha_i, \zeta_i) = (1/2, 1/2)$, $(\alpha_i, \zeta_i) = (1, 0)$ o $(\alpha_i, \zeta_i) = (0, 1)$. Las últimas dos opciones para la selección de estos parámetros fueron utilizadas en [6] en el diseño de heurísticas para la reducción del número de bloques no nulos en el complemento de Schur. Para una cara interior f se tiene

$$\oint_f \widehat{u_h}_{\{f,T_k\}} \boldsymbol{\psi}_m^k \cdot \vec{\boldsymbol{n}}_k \;=\; \alpha_i \oint_f u_h^k \boldsymbol{\psi}_m^k \cdot \vec{\boldsymbol{n}}_k + \zeta_i \oint_f u_h^i \boldsymbol{\psi}_m^k \cdot \vec{\boldsymbol{n}}_k.$$

La primera integral involucra únicamente las trazas de u_h dentro de la celda T_k y contribuye al bloque B_{kk}. La segunda integral involucra las trazas por ambos lados de la cara y contribuye exclusivamente al bloque B_{ki} que se encuentra fuera de la diagonal.

La representación matricial correspondiente es:

$$\left[\alpha_i \oint_f u_h^k \boldsymbol{\psi}_m^k \cdot \vec{\boldsymbol{n}}_k\right] \;=\; \alpha_i \vec{\boldsymbol{n}}_k \otimes [IF]_f \quad \text{y} \quad \left[\zeta_i \oint_f u_h^i \boldsymbol{\psi}_m^k \cdot \vec{\boldsymbol{n}}_k\right] \;=\; \zeta_i \vec{\boldsymbol{n}}_k \otimes [EF]_f,$$

donde $[IF]_f$, lo llamaremos operador interior de cara, y $[EF]_f$, operador exterior de cara. Estos son matrices de tamaño dim $\mathcal{V}_h(T_k)\times$dim $\mathcal{V}_h(T_k)$ y dim $\mathcal{V}_h(T_k) \times$ dim $\mathcal{V}_h(T_i)$ respectivamente y, están definidas de la siguiente manera

$$([IF]_f)_{mn} = \oint_f \varphi_n^k \varphi_m^k \quad \text{y} \quad ([EF]_f)_{mn} = \oint_f \varphi_n^i \varphi_m^k. \tag{9}$$

De acuerdo al Cuadro 1 se tiene $\widehat{u_h}_{\{f,T_k\}} = g_{\mathcal{D}}$ para cualquier cara de borde f donde se imponen condiciones de Dirichlet. Por lo tanto f no

afecta la matriz B. Por otro lado, si se tienen condiciones de tipo Neumann o Robin en la cara f se tiene $\widehat{u_h}_{\{f,T_k\}} = u_h$, por lo que se puede fijar $\alpha_i = 1$ y $\zeta_i = 0$.

Como la malla consiste de celdas linealmente equivalentes, las integrales pueden ser todas precalculadas en la celda de referencia. Para el operador $[EF]_f$, las funciones de la base son evaluadas en ambos lados de la cara, por lo que se consideran todas las combinaciones posibles de pares de caras en la celda de referencia, así como sus rotaciones, para un total de 48 posibilidades. En la Figura 1 se presentan todas las posibles rotaciones para la cara sombreada. Dentro de la celda superior, la cara corresponde al lado 3, sin embargo dentro de la celda inferior esta cara puede ser etiquetada de cuatro formas distintas.

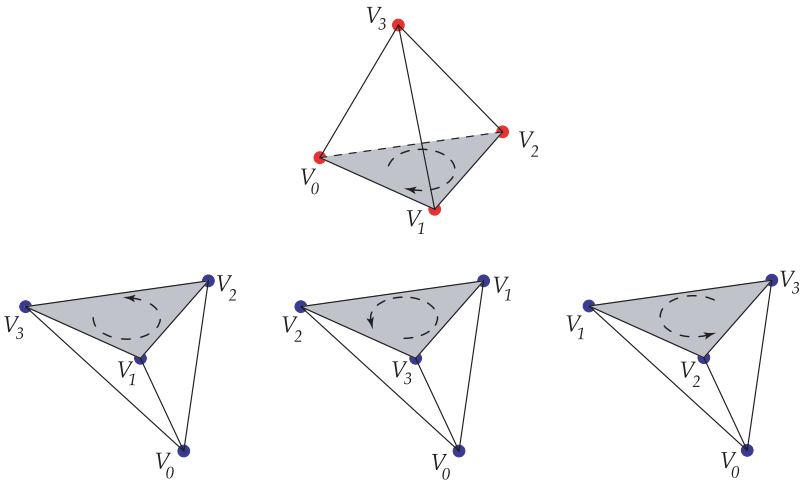

Figura 1: Posibles rotaciones de una cara compartida.

3.3 Operador de estabilidad

De acuerdo con [7] el parámetro de estabilidad $\alpha_f(\cdot)$ se define como sigue

$$\alpha_f(u_h) \;=\; \eta_f \left(u_h^k \vec{\boldsymbol{n}}_k + u_h^i \vec{\boldsymbol{n}}_i \right) \;=\; \eta_f \left(u_h^k - u_h^i \right) \vec{\boldsymbol{n}}_k.$$

De donde se tiene que la matriz de estabilidad S es una matriz esparcida de $N \times N$ bloques, donde los bloques S_{kk} y S_{ki} se leen

$$S_{kk} \;=\; \sum_{f \in \partial T_k} \eta_f [IF]_f \quad \text{y} \quad S_{ki} \;=\; -\eta_f [EF]_f,$$

cuando f es una cara interior. Para una cara de borde f con condiciones de Dirichlet, esta contribuye al bloque S_{kk} de igual manera a la anterior.

Cuando a f se le imponen condiciones de Neumann, f no afecta la matriz S. Para f con condiciones de Robin, se tiene $\eta_f = a$ y f contribuye al bloque S_{kk} análogamente a una cara interior.

3.4 Vector de la derecha

El vector global del sistema lineal (7) se descompone en dos vectores: el primero, b_q, se obtiene a partir de las condiciones de frontera de tipo Dirichlet. Si $f \in \partial T_k \cap \partial\Omega_{\mathcal{D}}$, la contribución de la cara f a este vector b_q se obtiene por el vector local

$$\left[\oint_f g_{\mathcal{D}} \boldsymbol{\psi}_m^k \cdot \vec{\boldsymbol{n}}_k\right] \;=\; \vec{\boldsymbol{n}}_k \otimes \left[\oint_f \varphi_m^k g_{\mathcal{D}}\right].$$

El tamaño de este vector es igual a 3dim $\mathcal{V}_h(T_k)$. La segunda componente es el vector b_u que depende del término fuente f, así como de los tres tipos de condiciones de frontera. La contribución final de la celda T_k al vector b_u es un vector de dimensión dim $\mathcal{V}_h(T_k)$ el cual puede ser calculado de la siguiente manera:

$$\left[\int_{T_k} f\,\varphi_m^k\right] + \sum_{f\in\partial T_k\cap\partial\Omega_{\mathcal{D}}} \left[\eta_f \oint_f g_{\mathcal{D}}\varphi_m^k\right] + \\ + \sum_{f\in\partial T_k\cap\partial\Omega_{\mathcal{N}}} \left[\oint_f g_{\mathcal{N}}\varphi_m^k\right] + \sum_{f\in\partial T_k\cap\partial\Omega_{\mathcal{R}}} \left[\oint_f g_{\mathcal{R}}\varphi_m^k\right].$$

El primer término se debe a la fuente, función f, el segundo, tercer y cuarto término corresponden a condiciones de frontera de tipo Dirichlet, Neumann y Robin, respectivamente.

3.5 Estructura de datos

Para el manejo de matrices D, B S y $\mathcal{A}$ utilizamos una versión por bloques del formato "Compressed Sparse Row ", o CSR [11], el cual se utiliza comúnmente para el almacenamiento de matrices esparcidas. Esta versión es ideal para la versión p del método: por un lado se asume una estructura de bloques la cual es natural para aproximaciones de alto orden y por otro, se almacena solamente los bloques no nulos de la matriz utilizando una estrategia similar a la del formato CSR. En lugar de utilizar librerías existentes, como por ejemplo la librería BPKIT desarrollada en [8], se implementó una versión más modesta, enfocada a las necesidades del método.

Para la implementación de los operadores numéricos se consideraron dos estructuras. La clase **Operator2D** almacena exclusivamente los operadores que requieren integrales de superficie, por ejemplo $[IF]$, $[EF]$ y los

operadores para las condiciones de frontera. La clase **Operator3D** contiene los operadores que se construyen a partir de integrales de volumen, por ejemplo la matriz de masa local y su inversa, y los operadores diferenciales DX,DY y DZ. El método *setData* inicializa las tablas internas en ambas clases y requiere de un apuntador para la interfaz de bases de polinomios **Base3D**; y de un apuntador para la interfaz de cuadraturas en 2D **Quad2D** o **Quad3D** en 3D. De esta forma el código no depende ni de las bases de polinomios ni de las cuadraturas.

En el siguiente fragmento de código ilustramos ambas estructuras de datos. Para obtener el operador $[IF]$ de una cara se utiliza el método *Operador2D::setIF()* este requiere como parámetro de entrada el índice local de la cara. En cambio para obtener el operador $[EF]$ se utiliza el método *Operador2D::setEF()*. Este necesita tres párametros de entrada: los índices locales de la cara y el índice de la rotación (ver anexo).

```
class Operator2D {
public:
  Operator2D();
 ~Operator2D();
  void   setData(const Basis3D*,
                 const Quad2D*);
  Block* getIF(uint);
  Block* getEF(uint, uint, uint);
  void   getProjection(const Tetra*,
                       uint, uint,
                       Function3D,
                       double*);
private:
   ...
};


class Operator3D {
public:
  Operator3D();
 ~Operator3D();
  void   setData(const Basis3D*,
                 const Quad3D*);
  Block* getMassMatrixInv();
  Block* getDX();
  Block* getDY();
  Block* getDZ();
  void   getProjection(const Tetra*,
                       Function3D,
                       double*);
private:
   ...
};
```

4 Validación del código

4.1 Convergencia del método LDG en 3D

A continuación se presentan algunos experimentos numéricos para validar la implementación descrita previamente, en especial, las tasas de convergencia y el comportamiento del condicionamiento espectral del complemento de Schur para varios grados de aproximación. En todos los ejemplos se utilizó el software TETGEN, desarrollado por H. Si, [12], para la generación de las mallas no estructuradas en 3D. En la siguiente figura mostramos una de las mallas generadas por TETGEN utilizadas en los experimentos numéricos.

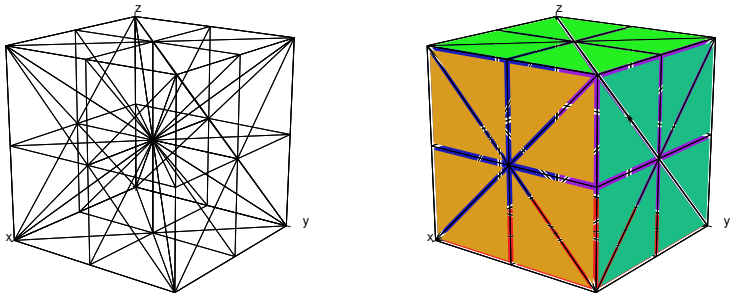

Figura 2: Ejemplo de malla (izquierda) y frontera (derecha).

4.1.1 Problema anisotrópico

En este ejemplo se considera el problema modelo (1) en el dominio $\Omega = (0,1)\times(0,1)\times(0,1)$ con condiciones de frontera de tipo Dirichlet y difusión anisotrópica. La matriz de difusión es

$$\mathcal{K} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}.$$

La función f se escoge de tal forma que la solución exacta sea la función

$$u(x,y,z) \;=\; \sin(\pi x)\sin(\pi y)\sin(\pi z).$$

En la Figura 3 mostramos la norma L_2 de los errores $u-u_h$ y $\nabla u - \nabla u_h$ al utilizar aproximaciones de grado $p = 1, 2, 3$. Mediante regresión lineal se

obtienen las pendientes 1.92, 3.00, 4.32 para $\| u - u_h \|$; y, 0.90, 1.92, 3.20 para $\| \nabla u - \nabla u_h \|$ con $p = 1, 2, 3$, respectivamente. Lo cual muestra un comportamiento asintótico de orden $\mathcal{O}\left(h^{p+1}\right)$ para el error en el potencial y de orden $\mathcal{O}\left(h^{p}\right)$ para el gradiente tal como lo predicen los estimados de error *a priori* propuestos en [7, 10].

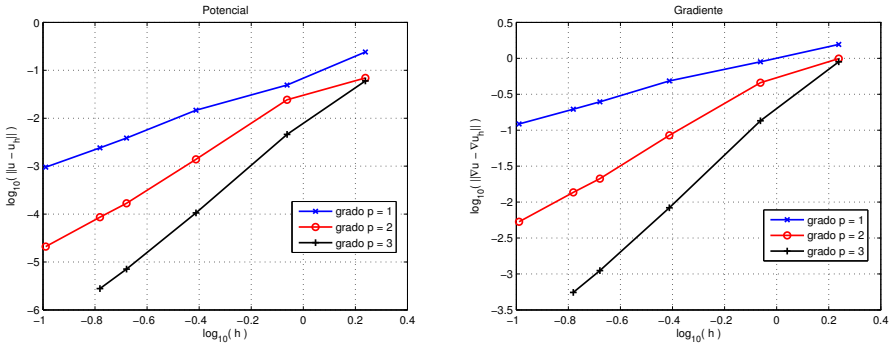

Figura 3: Error del potencial (izquierda) y del gradiente (derecha).

4.1.2 Condiciones de Dirichlet no cero

En este ejemplo consideramos el problema modelo con difusión isotrópica, $\mathcal{K} = I_d$ pero con condiciones de Dirichlet no cero. La función f se define de tal forma que la solución exacta sea la función

$$u(x, y, z) \;=\; e^{x+y+z}.$$

En la figura 4 podemos apreciar que la norma L_2 del error se comporta de manera asintótica como lo indica el estimado de error *a priori* propuesto en [7].

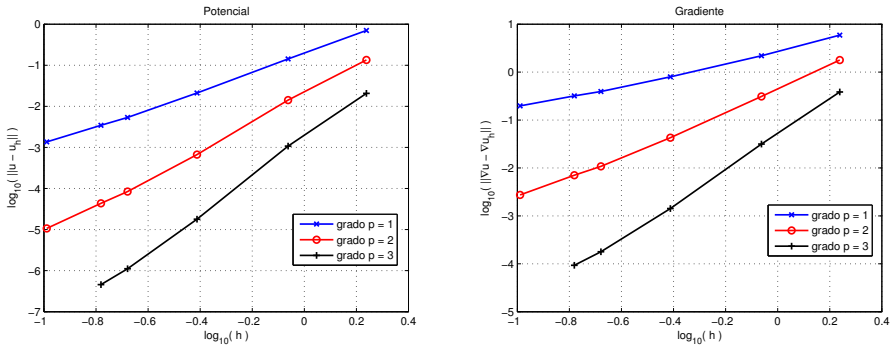

Figura 4: Error del potencial (izquierda) y del gradiente (derecha).

4.2 Condicionamiento espectral

En [5] se demostró que el condicionamiento espectral del complemento de Schur para el método LDG satisface la siguiente desigualdad

$$\kappa_2(\mathcal{A}) \;\leq\; C_1\,(C_2 + C_3\eta)\left(C_4 \max\left\{1, \frac{1}{\eta}\right\}\right) h^{-2}, \tag{10}$$

donde $C_i,\ i = 1, \ldots, 4$ son constantes positivas que no dependen de h. Esta cota fué comprobada numéricamente para discretizaciones con mallas no estructuradas en 2D. En este experimento mostramos que dicho comportamiento también se verifica para mallas no estructuradas en 3D. En la Figura 5 (izquierda) se muestra el comportamiento del condicionamiento con respecto al tamaño de la malla. Mediante regresión lineal se obtienen las siguientes pendientes -2.74, -2.27 y -2.11 para $p = 1, 2, 3$ respectivamente, lo cual muestra el comportamiento asintótico de orden $\mathcal{O}\left(h^{-2}\right)$ independientemente del grado de aproximación utilizado.

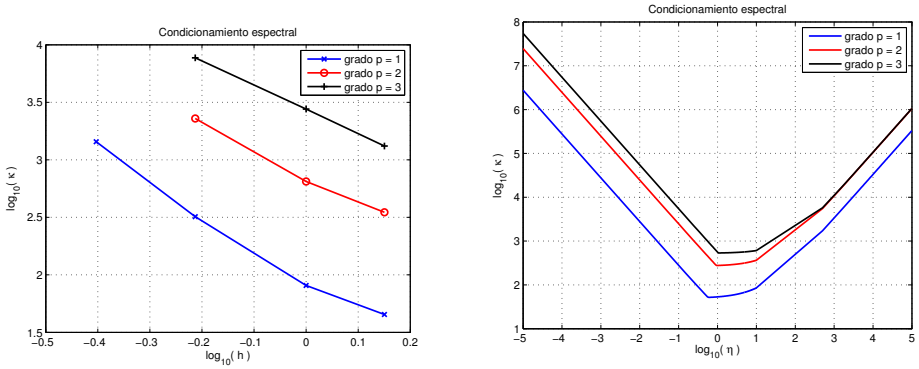

Figura 5: Condicionamiento espectral: $\kappa(h)$ (izquierda) y $\kappa(\eta)$ (derecha) para $p = 1, 2, 3$.

En la Figura 5 (derecha) mostramos el condicionamiento espectral κ como función del parámetro de estabilidad η. Se observa un comportamiento de $\mathcal{O}\,(1/\eta)$ para $\eta \ll 1$ y de $\mathcal{O}\,(\eta)$ para $\eta \gg 1$, lo que muestra que la cota propuesta en la desigualdad (10) también es precisa para mallas no estructuradas en 3D.

5 Conclusiones

En este artículo se han descrito algunas de las estructuras de datos más relevantes para una implementación eficiente del método LDG aplicada a problemas lineales elípticos, utilizando aproximaciones de alto orden y

mallas no estructuradas en 3D. Finalmente los experimentos numéricos presentados corroboran los estimados de error a priori y comportamiento del condicionamiento espectral en mallas no estructuradas en 3D.

Referencias

[1] Anderson, M.; Bai, Z.; Bischof, C:; Blacford, S.; Demmel, J.; Dongarra, J.; Du Croz, J.; Greenbaum, A.; Hammarling, S.; McKennes, A.; Sorensen, D. (1999) *LAPACK User's Guide*, 3rd edition. Society for Industrial and Applied Mathematics, Philadelphia PA.

[2] Arnold, D.N.; Brezzi, F.; Cockburn, B.; Marini, D. (2002) "A unified analysis for discontinuous Galerkin methods for elliptic problems", *SIAM J. Num. Anal.* **39**(5):1749–1779.

[3] W. Bangerth, R. Hartmann, and G. Kanschat. Deal.II: a general purpose object oriented finite element library. *ACM. Trans. Math. Software.*, 33(4):24:1–24:27, 2007.

[4] Bustinza, R.; Gatica. G.N. (2004) "A local discontinuous Galerkin method for nonlinear diffusion problems with mixed boundary conditions", *SIAM J. Sci. Comput.* **26**(1): 152–177.

[5] Castillo, P. (2002) "Performance of discontinuous Galerkin methods for elliptic PDE's", *SIAM J. Sci. Comput.* **24**(2): 524–547.

[6] Castillo, P. (2010) "Stencil reduction algorithms for the local discontinuous Galerkin method", *Internat. J. Numer. Methods Engrg.* **81**: 1475–1491.

[7] Castillo, P.; Cockburn, B.; Perugia, I.; Schötzau, D. (2000) "An a priori error analysis of the local discontinuous Galerkin method for elliptic problems", *SIAM J. Num. Anal.* **38**(5): 1676–1706.

[8] Chow, E.; Heroux, M.A. (1998) "An object-oriented framework for block preconditioning", *ACM Trans. Math. Softw.* **24**: 159–183.

[9] Cockburn, B.; Shu, C.W. (1998) The local discontinuous Galerkin method for time-dependent convection-diffusion systems", *SIAM J. Num. Anal.* **35**: 2440–2463.

[10] Perugia, I.; Schötzau, D. (2002) "An *hp* analysis of the local discontinuous Galerkin method for diffusion problems", *J. Scientific Computing.* **17**: 561–571.

[11] Saad, Y. (2003) *Iterative Methods for Sparse Linear Systems*, 2nd edition. Society for Industrial and Applied Mathematics, Philadelphia PA..

[12] Tetgen, H.S. (2004) "A quality tetrahedral mesh generator and three dimensional Delaunay triangulator, v.1.3, user's manual". Technical Report 9, Weierstrass Institute for Applied Analysis and Stochastics, Berlin.

Anexo: Estructuras de datos

En esta sección se describen las estructuras de datos y algoritmos más relevantes para nuestra implementación del método LDG en mallas no estructuradas.

Matrices esparcidas por bloques

Los operadores discretos difusión (D), gradiente (B) y operador de estabilidad (S) para el LDG, son matrices esparcidas por bloques, donde los bloques pueden también ser esparcidos.

Para la implementación de una estructura de matrices espacida por bloques, recurrimos al formato CSR (Compressed Matrix Row), así se puede construir la matriz en formato CSR pero que en lugar de entradas en $\mathbb{R}$, las entradas sean bloques de tamaño $m \times n$ densos o en formato CSR. El formato CSR optimiza el tiempo en cálculos además del uso de memoria para almacenar la matriz.

Supongamos que la matriz A es esparcida por bloques, que tiene $M \times N$ bloques, y que las dimensiones del bloque B_{ij} son $m_i \times n_j$, es decir, los bloques pueden ser de tamaños no uniformes. Para construimos esta matriz en formato CSR, se verifica que el bloque B_{ij} no es el bloque nulo, a diferencia de antes, que se verificaba que el elemento no fuera cero.

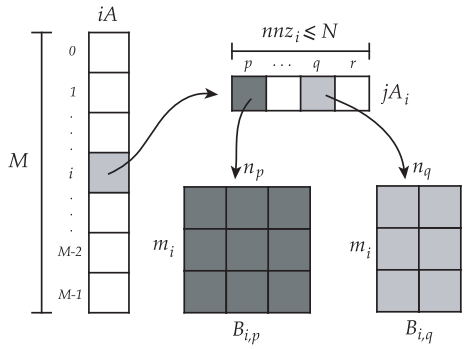

Figura 6: Estructura de datos para una matriz esparcida por bloques.

Para esta estructura definimos un arreglo iA de tamaño M el cual representa a las filas de la matriz A. Los campos de este arreglo apuntan a otro arreglo jA_i el cual contiene los bloques no nulos en la fila i.