

SÜREKSİZ GALERKİN SIKIŞTIRILABİLİR AKIŞLAR İÇİN GRAFİK İŞLEMCİ ÜNİTESİ HIZLANDIRILMIŞ ÇÖZÜCÜ

Sıla Yüksel* ve Ali Karakuş†
Orta Doğu Teknik Üniversitesi,
Ankara

ÖZET

Bu çalışmada süreksiz, sıkıştırılabilir akışlar için grafik işlemci ünitesi (GPU) hızlandırılmış, zamana bağlı bir Navier Stokes çözücüsü geliştirmek hedeflenmiştir. Navier Stokes denklemlerinin çözümünde uzay ayrıklaştırması için noktasal süreksiz Galerkin yöntemi ve denklemlerin içinde bulunan difüzyon terimleri için Symmetric Interior Penalty discontinuous Galerkin (SIPDG) [Arnold, 1982; Arnold, Brezzi, Cockburn ve Marini, 2002] metodu kullanılmıştır. Zaman ayrıklaştırması için düşük depolamalı, 4. sıra, 5 aşamalı, açık Runge-Kutta yöntemi [Kennedy, Carpenter ve Lewis, 2000] kullanılmıştır. Geliştirilen sayısal şemayı hızlandırmak için modern grafik işlemci üniteleri ve çok çekirdekli merkezi işlem birimleri kullanılmıştır. OCCA (open concurrent compute abstraction) kullanılarak hangi hesaplama aracının kullanılacağına çalışma sırasında karar verilmesi sağlanmış ve bu sayede çözücü platformdan bağımsız hale getirilmiştir. Düşük Mach sayılı bir akış problemi öncül testlerde kullanılmış ve bu testlerden elde edilen sonuçlar, uygulamalar ve değerlendirme bölümünde gösterilmiştir.

GİRİŞ

Süreksiz Galerkin yöntemi (DGM), yüksek dereceli bir sonlu elemanlar yöntemidir. DGM tamamen süreksiz, parçalı polinom yaklaşımını kullanır. Galerkin yöntemi çözüme adapte uzaysal ağ sıklaştırma ve seyreltme kolaylığı sağlar [Karakuş, Warburton, Aksel and Sert, 2016]. Metodun karmaşık geometrilere ve yüksek dereceli problemlere uygulanabilirliği, kısmi diferansiyel denklemlerin ayrıklaştırılması için en yaygın kullanılan sonlu hacim yöntemi gibi yöntemlerden daha fazladır ve diğer yöntemlere göre daha doğru sonuçlar sağlar [Shu, 2009; Hesthaven ve Warburton, 2007]. Yöntem, yüksek doğruluğun gerekli olduğu büyük ölçekli zamana bağlı hesaplamalarda çok etkilidir. Süreksiz Galerkin'in yüksek dereceli yapısı, temsil edilen dalga boyu başına daha az veri noktası ve dolayısıyla daha az bellek erişimi gerektirmesine olanak tanır. Genel sonlu eleman yöntemlerinin aksine bu yöntemle elde edilen denklemler, hesap yapılan hücreye özeldir. Bu nedenle, her hücre yalnızca komşularından bazı sınır verileri elde etmesi gereken ayrı bir eleman

*Makina Müh. Böl., E-posta: sila.yuksel@metu.edu.tr

†Doktor Öğretim Üyesi, Makina Müh. Böl., E-posta: akarakuş@metu.edu.tr

olarak düşünülebilir. Bu heterojen yapı yöntemde kesinlik kaybı olmaksızın, hesaplama sırasında, eleman topolojisi, yaklaşıklık derecesi ve hatta ana denklemlerin seçimi gibi özelliklerin elemandan elemana ve zaman içinde değişmesine olanak tanır. Kompakt formu ve zayıf haberleşme yapısı sayesinde paralel bilgisayar platformları için çok uygundur [Baggag, Atkins ve Keyes, 1999; Swirydowicz, Chalmers, Karakus and Warburton, 2019]. Paralel programlama, hız ve doğruluğun gerekli olduğu büyük ölçekli projeler için kritik öneme sahiptir. Tamamlanması gereken görevler birden çok bilgisayar veya birden çok çekirdek kullanılarak aynı anda çalıştırılabilmesi için paralelleştirilir. Bu sayede hem zamandan kazanılır hem de bilgisayarların daha verimli çalışması ve daha karmaşık sorunları çözebilmesi sağlanır. Son yıllarda dünya çapında kullanılmaya başlanan yeni nesil paralel programlama sistemlerinde ekran kartları (GPU) gibi çoklu hesaplama noktaları içeren hızlandırıcı mimarisi kullanılmaktadır. Bu değişiklik sayesinde, sayısal hesap başına düşen enerji gereksiniminde büyük oranda düşüş ve hesaplama gücünde artış sağlanmaktadır.

Platformdan bağımsız olan bu çözücü OCCA dili kullanılarak elde edilmiştir. OCCA, GPU ve CPU gibi farklı türdeki cihazları içeren bir ortamda programlamayı kolaylaştıran ve kullanıcının bir çalışma zamanında seçim yapmasına izin veren açık kaynaklı bir kütüphanedir. C, C++, Python gibi birçok programlama dilini destekler ve bunları tek bir Kernel dilinde özetler [Kirby ve Mavriplis, 2020]. Kod, libParanumal [Chalmers, Karakus, Austin, Swirydowicz ve Warburton, 2022] kitaplığı kullanılarak yazılmıştır. Yüksek dereceli ayrıklaştırmalar için yüksek performanslı çekirdekler, ölçeklenebilir bir kitaplık olan libParanumal tarafından sağlanır.

YÖNTEM

Sıkıştırılabilir Navier Stokes Denklemleri

Sıkıştırılabilir, kararsız ve viskoz akışta kütle, momentum ve enerji korunumu Navier Stokes denklemleriyle modellenebilir. Bu denklemler şu şekilde yazılabilir [Xia, Luo, Frisbey ve Nourgaliev, 2014]:

$$\frac{\partial \mathbf{U}(x, t)}{\partial t} + \frac{\partial \mathbf{F}_k(\mathbf{U}(x, t))}{\partial x_k} = \frac{\partial \mathbf{G}_k(\mathbf{U}(x, t), \nabla \mathbf{U}(x, t))}{\partial x_k} \quad (1)$$

U, F, G sırasıyla korunmuş değişkenler vektörü, advektif akı vektörü ve viskoz akı vektörünü temsil eder ve bu vektörler şu şekilde gösterilir:

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \end{pmatrix} \quad \mathbf{F}_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p \delta_{ij} \\ u_j(\rho e + p) \end{pmatrix} \quad \mathbf{G}_j = \begin{pmatrix} 0 \\ \tau_{ij} \\ u_i \tau_{ij} - q_j \end{pmatrix} \quad (2)$$

ρ yoğunluğu, u_i bir koordinat yönündeki (x_i) akış hızını, p basıncı ve e spesifik toplam enerjiyi temsil etmektedir. Bunlar haricinde bir de hal denklemine ihtiyaç duyulmaktadır. Bu denklem ise

$$p = (\gamma - 1)\rho \left(e - \frac{1}{2} u_i u_i \right) \quad (3)$$

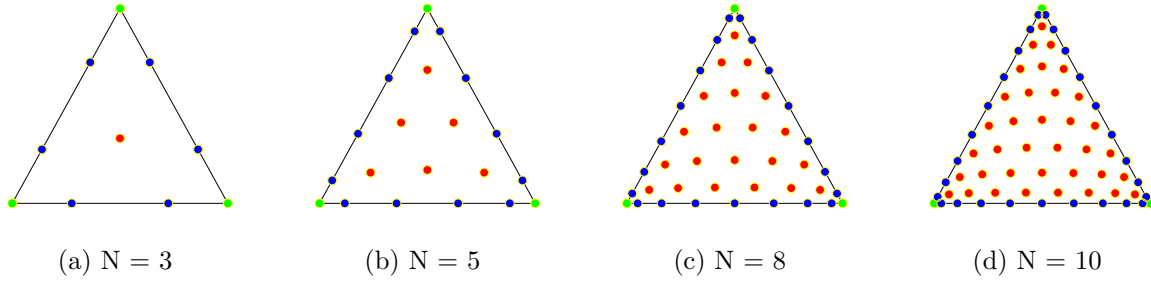
şeklinde gösterilir. Hava için $\gamma = 1.4$ olarak alınabilir. Viskoz stres tensörü τ_{ij} ve ısı akı vektörü q_j

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij} \quad q_j = \frac{1}{\gamma - 1} \frac{\mu}{Pr} \frac{\partial T}{\partial x_j} \quad (4)$$

şeklinde gösterilebilir. T sıcaklığı, Pr Prandtl sayısını (hava için 0.7 alınabilir) ifade eder.

Uzay ve Zaman Ayırıklaştırmaları

Eleman yöntemlerinin başarılı olabilmesi için fonksiyon enterpolasyonları yüksek kalitede olmalıdır. Bu kalite, uygun ortogonal temel fonksiyonlar ve uygun enterpolasyon noktaları ile sağlanır. Bu eleman yöntemleri, dörtgen/ altı yüzlü elemanlarda, Legendre polinomlarına ve bir boyutlu Gauss-Lobatto-Legendre (GLL) noktalarının tensör çarpımına dayalı bir düğüm yaklaşımı kullanır. Ancak üçgen/ 4 kenarlı elemanlar tensörsel olmadığı için, bu düğüm yaklaşımı kullanılamaz [Pasquetti ve Rapetti, 2006]. Bu problemde üçgen düğüm dağılımı kullanılmıştır ve düğüm yerleri, Çarpıtma ve Karıştırma düğümleri (Warp and Blend nodes [Warburton, 2006]) metodu kullanılarak belirlenmiştir. Üçgen eleman üzerindeki düğüm dağılımları Şekil 1'de gösterilmiştir.



Şekil 1: Bir eşkenar üçgen üzerinde değişen yaklaşıklık sırası için Çarpıtma ve Karıştırma düğümleri (Warp and Blend nodes).

Bu metoda göre N dereceli Lagrange polinomlarının minimum boyutu,

$$\dim P_N = N_p = \binom{N+2}{2} = \frac{(N+1)(N+2)}{2} \quad (5)$$

şekindedir. N_p , N dereceli Pascal üçgenindeki eleman sayısını belirtir.

Çarpıtma ve karıştırma üçgen düğüm dağılımında her yüzde N+1 düğüm (N-1 iç yüz düğümü ve komşu yüzlerle paylaşılan 3 tepe düğümü) toplamda 3(N+1) yüz düğümü vardır.

Sıkıştırılabilir Navier Stokes denklemleri süresiz Galerkin metodu ile şu şekilde ayrıklaştırılabilir.

$$\frac{d}{dt} \int_{\Omega_e} \mathbf{U} W d\Omega + \int_{\Gamma_e} \mathbf{F}_k(\mathbf{U}) \cdot \mathbf{n}_k W d\Gamma - \int_{\Omega_e} \mathbf{F}_k(\mathbf{U}) \cdot \frac{\partial W}{\partial x_k} d\Omega = \int_{\Gamma_e} \mathbf{G}_k(\mathbf{U}, \nabla \mathbf{U}) \cdot \mathbf{n}_k W d\Gamma - \int_{\Omega_e} \mathbf{G}_k(\mathbf{U}, \nabla \mathbf{U}) \cdot \frac{\partial W}{\partial x_k} d\Omega, \quad \forall W \in V^p \quad (6)$$

W test fonksiyonunu, Ω alanı, Ω_e alan içinde birbirleriyle örtüşmeyen elemanları, Γ_e elemanların sınırlarını, \mathbf{n}_k eleman sınırlarındaki birim dışa doğru normal vektörünü ifade eder. V^p ise p dereceden küçük tüm Legendre polinomlarının, çarpıtma ve karıştırma düğümleri üzerinde enterpolasyonu ile elde edilmiş Lagrange tipi polinom uzayını temsil eder.

Runge-Kutta (RK) yöntemleri, sıradan diferansiyel denklemlerin (ODEs) çözümlerini elde etmek için kullanılan bir dizi ara aşamalı hesaplama yöntemidir. Zamana bağlı kısmi diferansiyel denklemlerin (PDEs) zaman ayrıklaştırması için RK yöntemleri kullanılır. Problemden kullanılan PDE'ye bağlı olarak, zaman ayrıklaştırması için çok büyük bir ODE sistemi gerekebilir. Bu da hesaplama için gereken aşama sayısını artırır. Standart RK yöntemleri için depolama için gereken alan, aşama hesaplamalarının sayısı ile orantılıdır. Daha fazla depolama alanı çözümün daha yavaş olması anlamına gelir. Öte yandan, düşük depolamalı açık RK (LSERK) yöntemleri, depolama alanını artırmadan aşama sayısını artırmaya izin verir. Aşama sayısı arttıkça kararlılık sınırı ve zaman adımı boyutu artar ve böylece ODE'leri ve / veya PDE'leri çözmek için gereken süre azalır.

Güçlü kararlılık koruma, düşük depolama, 4. düzen, 5 aşamalı, açık Runge-Kutta yöntemi, zaman ayrıklaştırması için kullanılmıştır. Runge Kutta denklemleri şu şekilde yazılabilir:

$$\begin{aligned} \frac{dU_h}{dt} &= R(U_h, t) \\ dU_i &= a_i dU_{i-1} + \delta t R(U_i, t^n + c_i \delta t), \quad i \in [1, 5] \\ U_i &= U_{i-1} b_i dU_i, \quad i \in [1, 5] \end{aligned} \quad (7)$$

Paralel Uygulama

Denklemleri çözmek için gereken hesaplamaları üç ana başlıkta ele alabiliriz; hacim çekirdekleri tarafından gerçekleştirilen hacim entegrasyonu, yüzey çekirdekleri tarafından gerçekleştirilen yüzey entegrasyonu ve güncelleme çekirdekleri tarafından gerçekleştirilen zaman adımı güncellemesi. Bu hesaplamalar aşağıda verilen genel formda gösterilebilir.

$$\underbrace{\frac{dU}{dt}}_{\text{Güncelleme çekirdeği}} = R(U_h, t) = \underbrace{V(U_h)}_{\text{Hacim çekirdeği}} + \underbrace{S(U_h^-, U_h^+)}_{\text{Yüzey çekirdeği}} \quad (8)$$

Bir çalışma grubu bir veya daha fazla öğenin integrallerini çekirdeklerdeki her bir entegrasyon düğümünün katkısını hesaplar. Hacim integralleri hacim çekirdeklerinde; yüzey integralleri yüzey çekirdeklerinde hesaplanır. Her eleman için, hacim ve yüzey integrallerinden gelen katkılar, referans eleman üzerinde tanımlanan bir yerel matrisin hacim terimleri ve yüzey akılarının bir vektörünün çarpımı ile temsil edilir. Güncelleme çekirdekleri zaman entegrasyonu adımını hesaplar.

Hacim çekirdeğin ilk işlemi, elemanın alan değişkenlerini global bellekten paylaşılan belleğe kopyalamaktır. Bu işlem için N_p iş ögesi gereklidir. Daha sonra, depolanan paylaşılan bellek değişkenleri, kübatür entegrasyon noktalarına enterpole edilir. Bu işlem, aynı enterpolasyon matrisini kullanan bir dizi alan değişkeni matris-vektör çarpımı gerektirir. Her iş ögesi, bellek çakışmalarını önlemek için vektörlerle enterpolasyon matrisinin yalnızca bir satırını çarpar. Ortaya çıkan düğüm değerleri ve geometrik veriler, gelecekteki hesaplamaların hızlı değerlendirilmesi için kayıt belleğinde saklanır. Çekirdekteki ikinci adım kayıt belleğinde saklanan önceden elde edilen değerleri kullanarak hacim akı terimlerinin hesaplanmasıdır. Hacim akı terimi, paylaşılan bellek vektöründe saklanır. Son olarak, üç enterpolasyon matrisi ve daha önce paylaşılan bellekte depolanmış üç vektör kullanılarak üç matris-vektör çarpımı ile hacim terimleri enterpolasyon düğümlerine enterpole edilir.

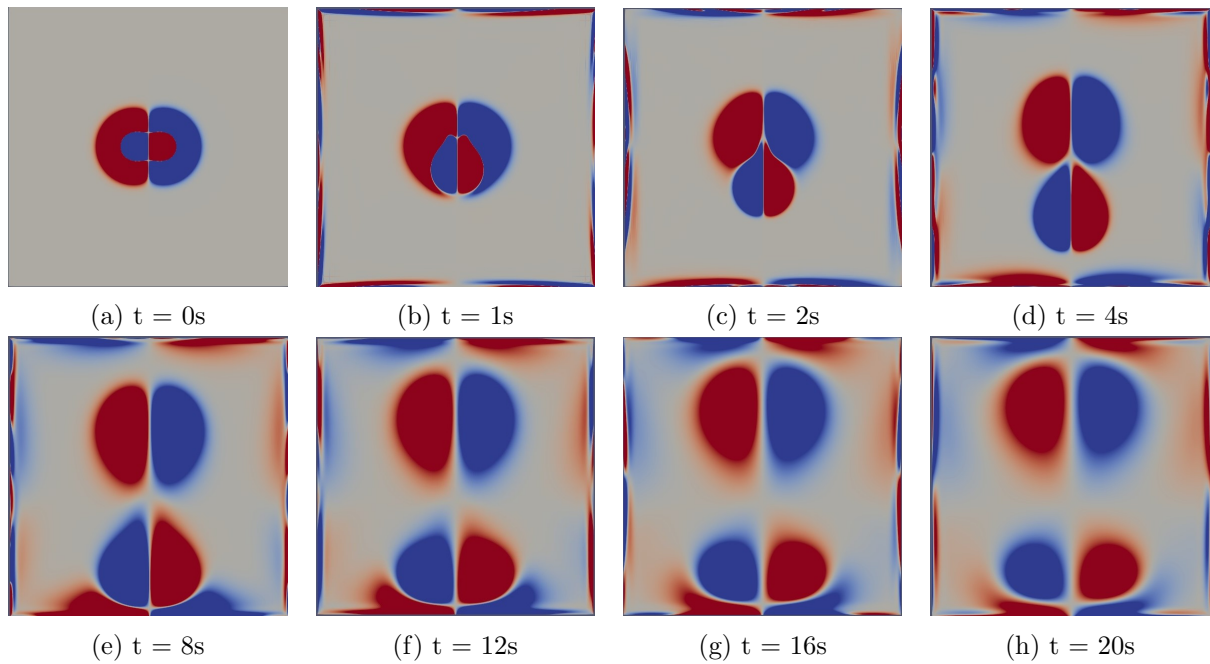
Yüzey çekirdeğinin ilk işlemi, elemanın alan değişkenlerini paylaşılan bellekte depolamaktır. İkinci işlem, standart düğüm değerlerinin yüzey kübasyon noktalarına enterpolasyonudur. Bu işlem, referans ögesinde tanımlanan aynı enterpolasyon matrisiyle önceden depolanmış paylaşılan vektörlerinin çarpımını içerir. Enterpolasyonlu değerler ve geometrik veriler kayıt belleğinde saklanır. Daha sonra, önceden elde edilen kayıt bellek değerleri kullanılarak, kübik düğümlerdeki akı fonksiyonu hesaplanır. Son olarak, matris - vektör çarpımı ile akı fonksiyonu enterpolasyon düğümlerine taşınır [Karakus, 2015].

UYGULAMALAR VE DEĞERLENDİRME

Girdap Dipol Akış Problemi

Öncül problem olarak düşük Mach sayılı, klasik bir girdap dipol (vortex-dipole) akış problemi çözülmüştür. Bu problemde 2 boyutlu, 4 kenarlı, 50x50 eleman kullanılmış, 7. dereceden bir polinom çözülmüştür. Problem izotermal kabul edilmiş ve viskozitesi 10^{-3} Pa.s alınmıştır. 20 saniye çalıştırılmıştır. Farklı zamanlardaki girdap görünümü Şekil 2'de verilmiştir. Şekildeki girdap kontur değerleri -0.1 ile 0.1 büyüklükleri arasındadır.

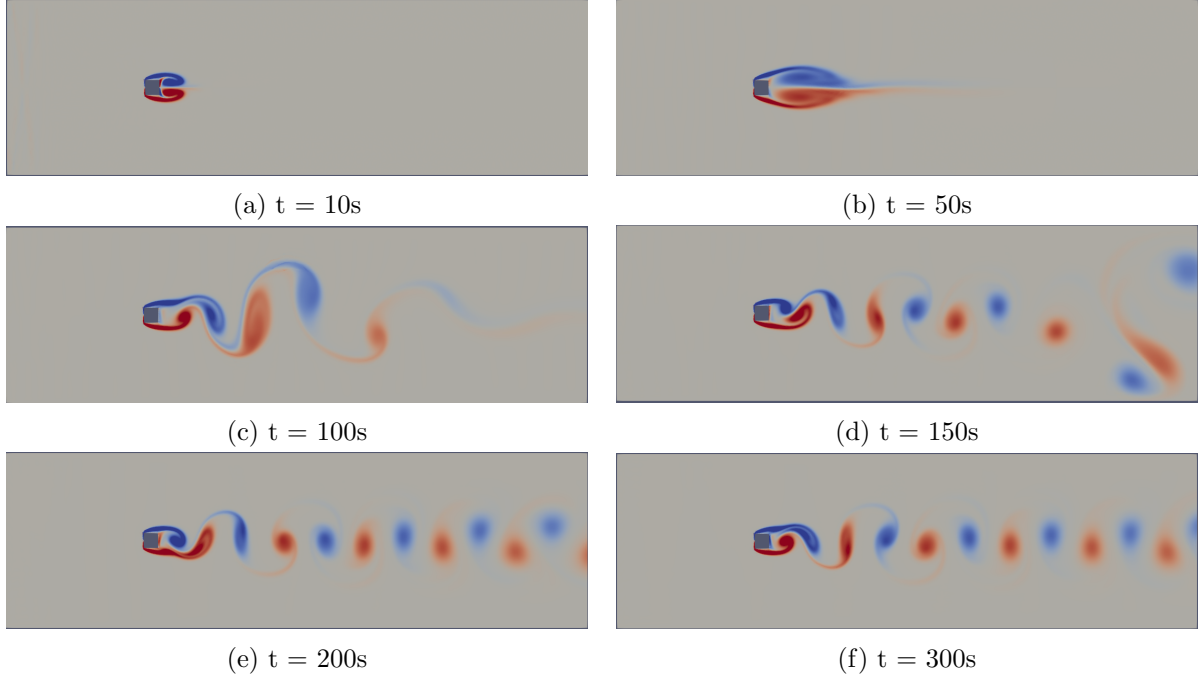
Girdap dipolleri eşit güçte ancak akış içinde farklı yönlere hareket eden iki girdaptan oluşur. Bu akışlardan biri pozitif girdaplı, diğeri negatif girdaplıdır ve bu nedenle toplam dolaşımı sıfırdır. Bu çemberin dışında girdap yoktur ve akış potansiyeldir. Viskoz akış sebebiyle girdap zaman geçtikçe geniş bir alana yayılır ve sonuç olarak dipolün gücü azalır, yavaşlar ve alanın merkezine doğru spiraller çizer. Navier-Stokes denklemi kullanılarak çözümler.



Şekil 2: Farklı zamanlardaki girdap görünümü.

Kare Silindir Üzerinde Akış Problemi

Klasik bir test problemi olan kare silindir üzerindeki 2 boyutlu düzgün akış problemi, çözücüyü test etmek için kullanılan bir başka problemdir. Bu problemde Mach sayısı 0.2, Reynolds sayısı 1000 olarak verilmiştir. Problem 300 saniye çalıştırılmıştır. Beklendiği üzere, kare silindirin akışın geldiği yöndeki köşelerinde akış ayrımı gerçekleşmiştir ve bu nedenle silindirden sonraki bölgede girdaplar oluşmuştur. Farklı zamanlardaki girdap görünümleri Şekil 3'te verilmiştir.



Şekil 3: Farklı zamanlardaki kare silindir arkasında oluşan girdap görünümü.

SONUÇ

Bu çalışmada süreksiz, sıkıştırılabilir akışlar için grafik işlemci ünitesi (GPU) hızlandırılmış, zamana bağlı bir Navier Stokes çözücüsü sunulmuştur. Uzay ayrıklaştırması için süreksiz Galerkin yöntemi ve üçgen düğüm dağılımı; zaman ayrıklaştırması için ise düşük depolama, 4. düzen, 5 aşamalı, açık Runge-Kutta yöntemi kullanılmıştır. Paralel programlama ile problem çözümü hızlandırılmıştır. 2 boyutlu, klasik test problemleri olan düşük Mach sayılı girdap dipol akış ve kare silindir üzerindeki düzgün akış problemleri çözücüyü test etmek için kullanılmış ve çıktı olarak zamana bağlı girdap görünümleri verilmiştir. Elde edilen sonuçlar, çözücünün düşük Mach sayılı akış problemlerinde doğru bir şekilde çalıştığını göstermektedir. Daha yüksek Mach sayılı akış problemlerinin çözülebilmesi için stabilizasyon metotları gereklidir. Bu çalışmanın bir sonraki aşamasında stabilizasyon metotlarını da libParanomal kütüphanesine ekleyerek sıkıştırılabilir akışlar için tüm Mach sayılarında doğru sonuçları alabilmek hedeflenmektedir.

Kaynaklar

- Arnold D., 1982. *An interior penalty finite element method with discontinuous elements.*, SIAM Journal on Numerical Analysis, 19(4): 742-760
- Arnold, D., Brezzi, F., Cockburn, B. and Marini, L. D., 2002. *Unified analysis of discontinuous Galerkin methods for elliptic problems.*, SIAM Journal on Numerical Analysis, 39(5): 1749-1779
- Baggag A, Atkins H., and Keyes D., 1999. *Parallel implementation of the discontinuous galerkin method.*, Technical report.
- Chalmers N., Karakus A., Austin A. P., Swirydowicz K., and Warburton T., 2022. *libParanumal: a performance portable high-order finite element library.*, URL <https://github.com/paranumal/libparanumal>. Release 0.5.0.
- Hesthaven J. S. and Warburton T., 2007. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications.*, Springer Science & Business Media.
- Karakus A., 2015. *GPU accelerated high-order discontinuous galerkin level set methods for incompressible multiphase flows.*, Middle East Technical University
- Karakus A., Warburton T., Aksel M.H. and Sert C., 2016. *A GPU-accelerated adaptive discontinuous Galerkin method for level set equation.*, International Journal of Computational Fluid Dynamics, 30(1): 56-68.
- Kennedy C. A., Carpenter M. H., and Lewis R. M., 2000. *Low-storage, explicit runge-kutta schemes for the compressible navier-stokes equations.*, Applied numerical mathematics, 35(3):177-219
- Kirby A. C. and Mavriplis D. J., 2020. *Gpu-accelerated discontinuous galerkin methods: 30x speedup on 345 billion unknowns.* In 2020 IEEE High Performance Extreme Computing Conference (HPEC), pages 1-7.
- Pasquetti R. and Rapetti F., 2006 *Spectral element methods on unstructured meshes: comparisons and recent advances.*, Journal of Scientific Computing, 27(1):377-387.
- Shu C.-W., 2009 *Discontinuous galerkin methods: general approach and stability.*, Numerical solutions of partial differential equations, 201.
- Swirydowicz K., Chalmers N., Karakus A. and Warburton T., 2019 *Acceleration of tensor-product operations for high-order finite element methods.*, The International Journal of High Performance Computing Applications, 33(4): 735-757.
- Warburton T., 2006. *An explicit construction of interpolation nodes on the simplex.*, Engineering Mathematics, 56:247-262.
- Xia Y., Luo H., Frisbey M., and Nourgaliev R., 2014. *A set of parallel, implicit methods for a reconstructed discontinuous galerkin method for compressible flows on 3d hybrid grids.*, Computers & Fluids, 98:134-151.