

Hücre Yapısının, Çok Hücreli, Olay Temelli, Molekül Dinamiği Simülasyonuna Etkisi

Bilge Esra Soran¹

F. Erdoğan Sevilgen²

^{1,2}Bilişim Teknolojileri Enstitüsü, TÜBİTAK Marmara Araştırma Merkezi, Kocaeli

^{1,2}Bilgisayar Mühendisliği Bölümü, Gebze Yüksek Teknoloji Enstitüsü, Kocaeli

¹e-posta: bilgeesra@yahoo.com

²e-posta: sevilgen@gyte.edu.tr

Özetçe

Olay temelli molekül dinamiği simülasyonları, çok parçacıklı sistemlerdeki olayları ardışık olarak ele alarak, sistemin davranışları üzerinde çalışmayı sağlayan önemli bir tekniktir. Çok hücreli yaklaşım sayesinde, bir sonraki olayın bulunması için sadece belirli bir alan içindeki moleküllerin incelenmesi yeterlidir. Bir molekül için kendi içinde bulunduğu hücre ve bu hücrenin komşu hücrelerindeki moleküller incelenerek molekülün karışacağı bir sonraki olay hesaplanabilmektedir. Bu çalışmada, yeni bir hücre dizilimi kullanılarak, bir hücrenin komşu hücre sayısı azaltılmıştır. Böylece, çarpışma kontrolü için bakılan hesaplama alanının daraltılması ve çalışma süresinin kısaltılması sağlanmıştır. Yapılan deneyler sonunda, klasik olay temelli molekül dinamiği simülasyonlarına oranla yaklaşık %35 performans artışı elde edilmiştir.

1. Giriş

Molekül dinamiği simülasyonları, çok parçacıklı sistemlerin özellikleri ve davranışı hakkında bilgi veren bir tekniktir [1]. Kontrollü bir ortamdaki tek tek moleküllerin hareket ve çarpışmalarının simülasyonundan oluşur.

Böyle bir sistemin simülasyonu için kullanılan, olay temelli ve zaman adımlı olmak üzere, farklı iki yöntem vardır. Zaman adımlı simülasyonlarda moleküller belirli zaman aralıklarında güncellenirler. Herhangi bir olayın kaçırılmaması için bu zaman aralıklarının yeteri kadar küçük seçilmesi gerekir. Fakat, zaman aralıklarını çok küçük seçmek, gereksiz güncellemelere ve delayı ile çalışma zamanının artmasına neden olur. Bu nedenle zaman temelli bir yaklaşımda zaman aralıklarını belirlemek önemli bir problemdir. Zaman temelli algoritmaların en büyük avantajı, kolayca paralel işlemciler üzerinde çalıştırılabilmeleridir [1].

Olay temelli simülasyonlarda ise olası olaylar zaman sırasında işlenir ve moleküller sadece bir olay gerçekleştiğinde güncellenir [2]. Bir sonraki işlenecek olayın bulunması için her bir molekül ikilisi arasındaki olası olayın dikkate alınması gereklidir. Bir sonra işlenecek olay, genel olarak, her bir molekülün olası ilk olayının bulunması ve daha sonra bu olaylar arasından sistemde meydana gelecek ilk olayın tesbit edilmesi ile bulunur. Bir molekülün ilk olayının bulunması, diğer bütün moleküller ile çarpışma durumu göz önüne alındığında, N adet molekül için, $O(N)$ zaman alacaktır. Bu sebeple, her bir olayın işlenmesi sonrasında, sistemde gerçekleşecek ilk olayın bulunması $O(N^2)$ zaman almaktadır. Bu çalışma süresi, çok basit bir gözlem ile $O(N)^2$ 'e

düşmektedir. Bir olay sonrasında sadece olaya katılan moleküllerin olası ilk olayları değişmekte ve yeniden hesaplanması gerekmektedir. Yani her olayın işlenmesi $O(N)$ zaman almaktadır ki, bu tip simülasyonlarda molekül sayısı çok yüksek olduğu düşünüldüğünde hala oldukça yüksek bir maliyettir. Bu nedenle simülasyonu etkin bir şekilde çalıştırmak için çeşitli iyileştirmeler öne sürülmüştür.

Moleküller arası çarpışmanın birbirine yakın moleküller arasında gerçekleşmesi daha olasıdır [3, 4, 5]. Ortamdaki molekül sayısı yüksek olduğunda, birbirine uzak moleküllerin çarpışma ihtimali çok daha düşüktür ve bunların dikkate alınması zaman kaybıdır. Bu nedenle bir komşuluk kavramı geliştirilmiştir [5]. Bu komşuluğun sınırlarını belirlemek üzere çok hücreli parçalama yöntemi kullanılmıştır. Çok hücreli yöntemde ortamın hücrelerden oluştuğu ve bu hücrelerin her birinin bir miktar molekül içerdiği düşünülür. Yapılan çalışmalarda, genel olarak eşit büyüklükte dikdörtgenler prizması şeklindeki hücreler kullanılmıştır [3, 4, 5]. Bu durumda, bir molekülün olası ilk olayının bulunması için, molekülün içinde bulunduğu hücre ve bu hücrenin komşu hücrelerinde yer alan moleküllerin dikkate alınması yeterlidir. Bu sayede çarpışma kontrolünün yapılacağı alan darlatılır ve büyük bir performans artışı sağlanır. Bir hücre içerisinde bulunan toplam molekül sayısı sabit bir değerde tutulabilirse bir molekülün ilk olası olayını sabit zamanda hesaplamak mümkündür. Rapaport çok hücreli yöntem kullanılması ile bir olayın işleme süresinin $O(\log N)$ 'e indirilebileceğini göstermiştir [5]. Buradaki logaritmik ifade, her bir molekülün olası ilk olaylarından faydalanarak sistemdeki ilk olayın hesaplanması için kullanılan, öncelik kuyruklarına ekleme ve çıkarma işlemlerinin çalışma zamanından kaynaklanmaktadır.

Bu çalışmada, olay temelli çok hücreli molekül dinamikleri simülasyonlarının performansının nasıl daha fazla artırılacağı sorusu üzerinde duruldu. Bu tip simülasyonlarda en çok zaman alan işlem bir molekülün olası ilk olayının hesaplanmasıdır. Bu işlemin çalışma süresi ise bir hücrenin içinde bulunan ortalama molekül sayısı ve hücrenin komşu hücrelerinin sayısının çarpımı ile doğru orantılıdır. Klasik çok hücreli yaklaşımda sınır hücreleri hariç her hücrenin 26 komşusu vardır. Yapılan farklı bir hücre dizilimi ile komşu sayısı 16'ya düşürülmüştür. Hücre başına düşen ortalama molekül sayısı değiştirilmeden, komşuluk sayısı azaltıldığı için, simülasyon performansında teorik olarak doğrusal bir artış (yaklaşık olarak %37) elde edilmesi beklenmektedir. Yapılan deneysel çalışma sonucunda, yaklaşık %35 oranında performans artışı elde edilmiştir. Beklenenden çok az da olsa daha düşük performans artışı

sağlanmasının sebebi yeni hücre yerleşimi sonucunda hücre geçişi olaylarındaki artıştır.

2. Molekül Dinamiği Simülasyonu

Molekül dinamiği simülasyonları, kontrollü bir ortamdaki moleküllerin davranışları hakkında deneysel çalışmalar yapılmasına olanak veren bir simülasyon tekniğidir. Simülasyon boyunca genellikle toplam enerji sabittir, fakat bazı çalışmalarda sıcaklık basınç gibi dış ortam özellikleri değiştirilerek de deneyler yapılmıştır [3]. Genellikle küresel olarak modellenen tek atomlu moleküller üzerinde çalışmalar yapılmış olsa da bir sistemde her molekül tek bir atomdan oluşmak zorunda değildir. Sistemde, moleküller arasında etkileşim olmadığı ve moleküllerin iki olay arasında sabit hızla hareket ettiği varsayılmaktadır. Simülasyon boyunca hareketli moleküller çarpışabilir hatta reaksiyona girebilirler. Fakat işlemleri basitleştirmek amacıyla, moleküller arası çarpışmaların elastik olduğu kabul edilmiştir.

Olay temelli bir molekül dinamiği simülasyonunda, moleküller sadece bir olay gerçekleştiğinde güncellenir. Moleküller arası çarpışma ve bir molekülün ortam duvarlarından birine çarpması olmak üzere, iki çeşit olay vardır. Sistemde gerçekleşecek ilk olayın bulunması için, her molekülün hem ortamın tüm duvarlarıyla hem de diğer tüm moleküllerle çarpışma kontrolü yapılmalıdır. Ortamda bulunan molekül sayısı N ise, bir molekül için olası ilk olayın hesabının maliyeti $O(d+N)$ 'dir. Burada d duvar sayısını ifade etmektedir ve genellikle simülasyon ortamı dikkörtgenler prizması olarak seçildiği için, 6'dır. Bu durumda çalışma süresi $O(N)$ 'dir. Bu maliyetin düşürülmesi ve simülasyon hızının arttırılması için uygulanabilecek önemli bir yöntem çok hücreli yaklaşımdır.

2.1. Çok Hücreli Yaklaşım

Bir molekül dinamiği simülasyonunda, gerçekleşmesi muhtemel tüm olaylar hesaplanmalıdır. Moleküller arası çarpışmalar ise genellikle birbirine yakın moleküller arasında gerçekleşir, uzak moleküller arasında çarpışma olasılığı oldukça düşüktür. Ortam moleküllerle dolu olduğu durumlarda çarpışması mümkün olmayan moleküllerin kontrol edilmesi oldukça fazla zaman kaybına yol açar. Bu yüzden, bir komşuluk kavramının geliştirildiği çok hücreli yaklaşımın kullanılması önerilmiştir [5]. Bu yaklaşımda, komşuluk sınırlarını çizebilmek için ortamın, her biri bazı molekülleri içeren hücrelerden oluştuğu düşünülür. Moleküller, ortamda serbestçe hareket eder ve hücreden hücreye geçebilirler. Bu durumu kontrol etmek için ise bundan sonra HÜCRE_GEÇİŞİ olarak bahsedilecek, yeni bir olay tipine ihtiyaç vardır. Sonuç olarak, bir çok hücreli olay temelli molekül dinamiği simülasyonunda, bir molekülün herhangi bir anda gerçekleştirebileceği üç farklı tip olay vardır:

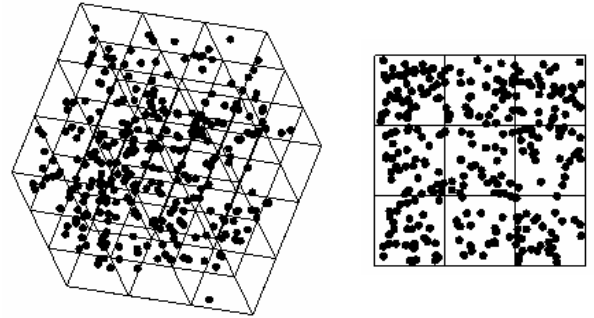
- MOLEKÜLLER_ARASI_ÇARPIŞMA,
- DUVARA_ÇARPIŞMA,
- HÜCRE_GEÇİŞİ.

Şekil 1'de klasik çok hücreli olay temelli molekül dinamiği simülasyonunun algoritması görülebilir.

1. Her molekülün ilk olayını hesapla.
2. Her hücrenin ilk olayını belirle.
3. Simülasyonun ilk olayını belirle.
4. Simülasyonun ilk olayını tetikle.
5. Katılan diğer moleküllerin sonraki olaylarını hesapla.
6. Katılan hücrelerin ilk olaylarını hesapla.
7. 3. adıma geri dön.

Şekil 1: Çok hücreli Olay Temelli Moleküller Dinamiği Simülasyonu Algoritması

Bir molekülün olası ilk olayının bulunması için, molekülün yapacağı ilk hücre geçişinin zamanı, yapacağı ilk molekül çarpışmasının zamanı ve ilk duvar çarpışmasının zamanı hesaplanmalıdır. Daha sonra bu üç zamandan en küçük olanı ve ilgili olay belirlenmelidir; bu olay molekülün olası ilk olayıdır. Molekülün olası ilk diğer bir molekül ile çarpışma olayının zamanı, molekülün içinde bulunduğu ev sahibi hücrenin ve ev sahibi hücrenin komşu hücrelerinin içindeki tüm moleküller kontrol edilerek hesaplanır. Böylece moleküller arası çarpışma hesabı için bakılacak molekül sayısı, klasik olay temelli yaklaşıma göre fark edilir şekilde azaltılır. Bununla birlikte, molekül yoğunluğu (birim alandaki molekül sayısı) çok düşük ise, bu yaklaşım simülasyonun toplam performansının azalmasına bile sebep olabilir. Çünkü, böyle bir sistemdeki bir molekül, gerçekten var olan bir MOLEKÜLLERARASI_ÇARPIŞMA olayı veya bir DUVARA_ÇARPIŞMA olayı gerçekleştirene kadar, bir çok HÜCRE_GEÇİŞİ olayı gerçekleştirebilir. Gerçekte olmayan ve çok hücreli yöntemin bir sonucu olan HÜCRE_GEÇİŞİ olayları ile uğraşmak simülasyonun performansını düşürür. Fakat molekül yoğunluğu yüksek simülasyonlarda, HÜCRE_GEÇİŞİ olaylarının gerçek olaylara oranı düşük olduğu için böyle bir problem ortaya çıkmamaktadır [3].



Şekil 2: Klasik çok hücreli yaklaşımla üç boyutlu ve iki boyutlu ortamın hücrelere bölünmesi – HücreDizilimi 1

Çok hücreli molekül dinamiği simülasyonlarında molekülleri içeren ortam genellikle bir dikkörtgensel prizmadır. Böyle bir ortamı hücrelere ayırıştırmak oldukça kolaydır (Bknz. Şekil 2). Bununla birlikte, “ortamın kaç hücreye bölünüşü” ve “her hücrenin kaç molekül içerdiğini” gibi değerler performans yönünden önemlidir. Önceki çalışmalarda, hücre başına molekül sayısının en uygun değerinin sekiz olduğu

gösterilmiştir [3]. Dolayısıyla, ortamın kaç hücreye bölüneceğine hücre başına düşecek molekül sayısı göz önünde bulundurularak karar verilebilir.

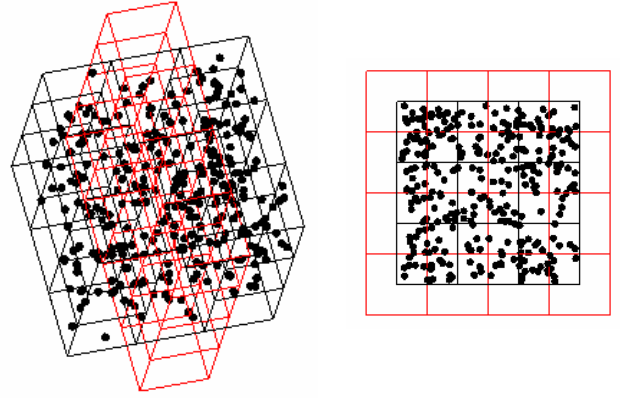
Sonuç olarak, çok hücreli yaklaşım kullanıldığında, muhtemel moleküller arası çarpışma hesaplama maliyetini $O(N)$ 'den $O((k+1)m)$ 'a düşer. Burada, m hücre başına düşen molekül sayısını ve k komşu hücre sayısını göstermektedir. Klasik çok hücreli yöntemde komşu hücre sayısının 26 ve hücre başına düşen molekül sayısının ortalama olarak 8 olduğu düşünüldüğünde bu maliyet $O(1)$ 'dir. Bir olay için harcanan toplam süreye bakıldığında ise, sabit olan bu maliyetin yanına sistemdeki ilk olayın hesaplanması için geçecek zaman eklenmelidir. Sistemdeki ilk olay, moleküllerin olası ilk olaylarının zamanları dikkate alınarak hesaplanacaktır. Bu hesaplama için, Rapaport ikili arama ağacı [5], Marín ise yığıt benzeri bir ikili ağaç [4] kullanmıştır. Her iki durumda da bir olayın gerçekleşmesi için gerekli ağaca ekleme ve çıkarma işlemleri logaritmik zaman almakta ve bir olayın işlenmesi için $O(\log N)$ zaman gerekmektedir.

3. Önerilen Hücre Dizilimi

Çok hücreli yaklaşım kullanıldığında, moleküller arası çarpışma olaylarının belirlenmesinde sağlanan hızlanma sayesinde simülasyon zamanı belirgin biçimde azalır. Çok hücreli yöntemde, her molekülün ilk çarpışacağı molekülü aranırken dikkate alınan arama alanı daraltılmış ve sadece molekülün içinde bulunduğu evsahibi hücre ve evsahibi hücrenin komşu hücreleri ile sınırlandırılmıştır. Hücrelerdeki ortalama molekül sayısı sabit kalmak koşulu ile bir hücreye komşu olan hücrelerin sayısı azaltılabilirse çalışma zamanında ilave bir iyileşme elde edilebilir.

Bu çalışmada, molekülün ilk olayının belirlenmesi adımında performans artışı sağlamak amacıyla yeni bir hücre dizilimi önerilmektedir. Bu dizilim, bir hücrenin komşu hücre sayısını azaltarak, performans artışı sağlarken, toplam kinetik enerjinin ve toplam momentumun korunmasını garanti etmektedir.

Önerilen hücre dizilimi Şekil 3 de verilmiştir. Bu dizilim, klasik hücre dizilim kullanılarak şöyle elde edilebilir: Ortamın z eksenini boyunca paralel dizilmiş (x ve y ekseninde sıralanmış hücrelerden oluşan) dilimlerden oluştuğunu düşünelim. Bu dilimlerden z eksenini boyunca, tek numaralı olanlar oynatılmadan, çift numaralı dilimler x ve y eksenleri boyunca hücre uzunluğunun yarısı kadar ötelensin. Bu durumda bazı hücreler ortamın dışına taşacaktır. Bu hücrelerdeki moleküllerin ortamın dışına çıkması, ortam duvarının yeri kontrol edilerek kolayca engellenebilir. Şekil 3'de de görüldüğü gibi her hücrenin kendi diliminde sekiz komşusu olduğu halde bir önceki ve bir sonraki dilimlerde komşu sayısı dörttür. Bu sayede bir hücrenin komşu hücre sayısı 26'dan 16'ya indirilmiştir.



Şekil 3: Üç boyutlu ortam için önerilen hücre diziliminin iki farklı açıdan gösterimi – HücreDizilimi2

4. Gerçekleme ve Algoritma Detayları

Veri Yapıları: Simülasyon sırasında, bir olay gerçekleştiğinden sonra, gerçekleştirilecek sonraki ilk olayın belirlenmesi gerekmektedir. Bu nedenle, hücrelerdeki moleküller, ilk olay zamanlarına göre sıralı tutulmuştur. Bu da, hücrenin içinde, bağlantılı liste şeklinde tasarlanmış öncelik kuyruğu veri yapısı kullanılarak sağlanmıştır. Hücreleri de, hücreler içinde oluşabilecek ilk olay zamanlarına göre sıralı tutmak için benzer bir öncelik kuyruğu kullanılmıştır.

Başlangıç Süreci: Başlangıç fazı boyunca hücrelerin pozisyonları, komşuları, bitişik ortam duvarları hesaplanmalıdır. Belirlenen molekül sayısına göre, ortamdaki yoğunluk homojen olacak şekilde, moleküllerin pozisyonlarına karar verilmelidir. Ayrıca, moleküllerin başlangıç vektörel hızları da belirlenmelidir.

Olay İşleme Süreci: Simülasyon ardışık olayların işlenmesiyle akar. Bir olayın hesaplanması demek, olay zamanının, olay tipinin, ilgili hücre var ise ilgili hücrenin, ilgili molekül var ise ilgili molekülün ve ilgili duvar var ise ilgili duvarın bulunması demektir. Molekülün merkezinden hücrenin bir sınırına olan uzaklığının, molekülün o yöndeki hızına bölünmesiyle molekülün bu hücre sınırından komşu hücreye geçiş zamanı hesaplanır. 3 boyutlu bir uzayda bir vektörün üç bileşeni olduğundan, sadece bu üç yöndeki hücre sınırları için kontrol etmek yeterli olacaktır. Muhtemel hücre geçiş zamanları hesaplandıktan sonra bunların en küçük olanı molekülün hücre geçiş zamanıdır. Olay hesaplama algoritması Şekil 4'de görülebilir.

Eğer bir molekül için herhangi bir gelecek olay bulunmazsa, olay tipi ve olay zamanı olarak -1 kaydedilir ve molekül öncelik kuyruğunun en sonuna yerleştirilir. Benzer durum hücreler için de geçerlidir.

1. Eğer molekül bir sınır hücresinde ise muhtemel duvar çarpışmalarını hesapla.
2. Molekülün içinde bulunduğu hücredeki moleküllere ve komşu hücrelerdeki moleküllere bakarak muhtemel molekül çarpışmalarını hesapla.
3. Molekülün içinde bulunduğu hücrenin sınırlarını kontrol ederek, muhtemel hücre geçişlerini hesapla.
4. En düşük pozitif olay zamanını molekülün ilk olayı olarak belirle.

Şekil 4: Olay hesaplama algoritması

Görselleştirme Süreci: Akışı grafiksel olarak takip edebilmek için, simülasyon OpenGL ve Glut kütüphaneleri kullanılarak görselleştirilmiştir.

5. Deneysel Çalışmalar

Tüm deneysel çalışmalar Pentium 4 2.0 GHz işlemci ve 1 GB RAM'e sahip bir bilgisayarda yapılmıştır. Yazılımlar C++ dili kullanılarak kodlanmıştır. Örnek problem olarak, 1000, 2000, 3000 ve 4000 molekül içeren sistemler seçilmiştir. Her bir örnek problem 500, 600, 700, 800, 900 ve 1000 simülasyon zaman birimi kadar çalıştırılmıştır. Yapılan deneylerde elde edilen sonuçlar Tablo 1, 2, 3, 4, ve 5 de özetlenmiştir. Tablo 1 tüm deneyler için **çarpışma sayılarını** gösterilmektedir. Klasik hücre dizilimi ile yapılan deneylerdeki **hücre geçişleri sayısı** Tablo 2 de ve saniye cinsinden **simülasyon zamanları** Tablo 4 de verilmiştir. Tablo 3 de önerilen hücre dizilimi ile yapılan deneylerde elde edilen **hücre geçişleri sayısı** özetlenmiştir. Aynı hücre dizilimi için, saniye cinsinden **simülasyon zamanları** Tablo 5 de açıklanmıştır.

Tablo 1: Çarpışma Sayıları

Molekül Sayısı	Simülasyon Birimleri					
	500	600	700	800	900	1000
1000	898	1053	1229	1382	1555	1731
2000	3801	4457	5131	5817	6521	7218
3000	8468	10016	11558	13078	14687	16194
4000	15029	17661	20398	23117	25901	28655

Tablo 2: Hücre Geçişleri Sayısı

Molekül Sayısı	Simülasyon Birimleri					
	500	600	700	800	900	1000
1000	19321	23117	26853	30618	34367	38076
2000	37939	45316	52612	59973	67292	74556
3000	56217	67152	78061	88983	99798	110799
4000	74463	88981	103598	118110	132651	147131

Tablo 3: Hücre Geçişleri Sayısı

Molekül Sayısı	Simülasyon Birimleri					
	500	600	700	800	900	1000
1000	20142	24126	28026	31918	35821	39693
2000	39547	47270	54963	62591	70215	77861
3000	58457	69881	81339	92771	104051	115438
4000	77702	92942	108131	123313	138493	153561

Tablo 4: Simülasyon Zamanları (sn)

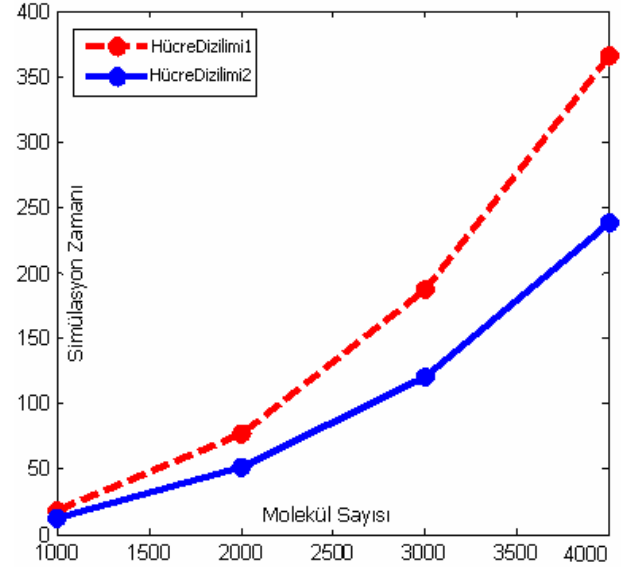
Molekül Sayısı	Simülasyon Birimleri					
	500	600	700	800	900	1000
1000	10	11	14	15	17	19
2000	39	48	53	61	68	77
3000	95	112	130	146	164	187
4000	204	226	257	298	324	366

Tablo 5: Simülasyon Zamanları (sn)

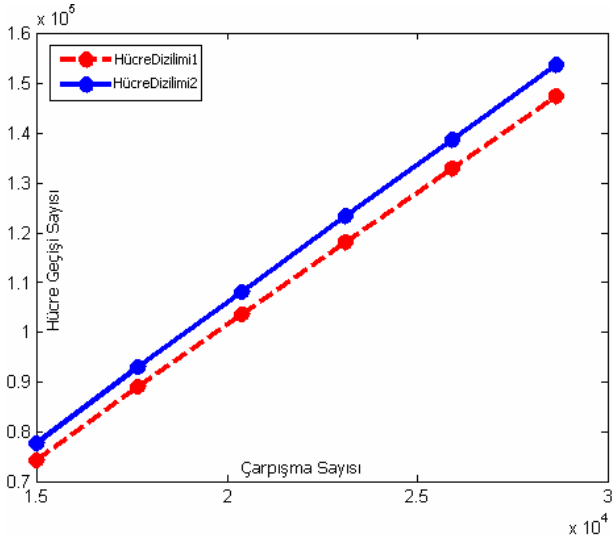
Molekül Sayısı	Simülasyon Birimleri					
	500	600	700	800	900	1000
1000	7	9	9	11	12	13
2000	27	32	37	43	47	52
3000	65	76	88	102	114	121
4000	124	151	167	192	211	238

Önerilen hücre diziliminin kullanılmasıyla hücre geçiş sayısının artışına rağmen, simülasyon performansının yaklaşık olarak %35 oranında arttığı açık olarak görülmektedir. Sonuç olarak, bir hücre içinde bulunan ortalama molekül sayısının her değeri için, komşuluk sayısının azaltılması performansı yaklaşık aynı miktarda artırmaktadır.

Elde edilen sonuçlara ait grafikler Şekil 4, 5 ve 6 da gösterilmiştir. Şekil 4 simülasyonun 1000 birim çalıştırılmasıyla elde edilen sonuçları özetlemektedir. HücreDizilimi2 için geçen simülasyon zamanlarının HücreDizilimi1 ile kıyaslandığında daha az olduğu görülmektedir. Burada HücreDizilimi1 geleneksel hücre tipini (Komşuluk sayısı 26'dır) ve HücreDizilimi2 ise değiştirilmiş hücre tipini (Komşuluk sayısı 16'dır) göstermektedir.

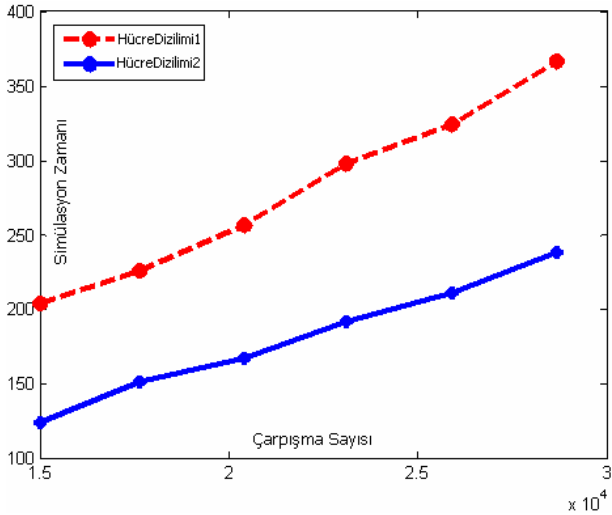


Şekil 5 simülasyonun 4000 molekül sayısı ile her iki hücre tipi için 500, 600, 700, 800, 900, 1000 birim çalıştırılmasıyla elde edilmiştir. Buradan değiştirilmiş hücre tipi ile hücre geçişleri sayısının arttığı görülmektedir. Bu bizim yaptığımız iyileştirmenin bir yan etkisidir fakat performansı çok fazla etkilememektedir. Bu artış sebebi ile beklenen yaklaşık %37'lik performans iyileşmesi %35 olarak gerçekleşmiştir.



Şekil 5: Hücre Geçiş Sayısı – Çarpışma Sayısı

Son olarak, Şekil 6 simülasyonun 4000 molekül sayısı ve her iki hücre tipi için 500, 600, 700, 800, 900, 1000 birim çalıştırılmasıyla elde edilmiş genel sonuçları göstermektedir. Değiştirilmiş hücre tipi ile birim zamanda daha çok çarpışma olayı işlenebilmektedir.



Şekil 6: Simülasyon Zamanı – Çarpışma Sayısı

6. Sonuç

Bu güne kadar, molekül dinamiği simülasyonlarındaki en büyük performans artışı çok hücreli yöntem sayesinde elde edilmiştir [3, 4, 5, 6, 7]. Bu çalışmada, çok hücreli yöntemi farklı hücre dizilimleriyle uyarılmanın klasik hücrelere ayırıştırma ile elde edilen performansı ciddi oranda arttırdığı gösterilmiştir. Önerilen hücre dizilimi sayesinde, bir molekül için dikkate alınacak hücre sayısı 27 den 17 ye indirilmiş yani $(27-17)/27*100 = \% 37$ kadar azaltılmıştır. Simülasyon hızının ise Tablo 4 ve Tablo 5 deki işaretli veriler karşılaştırıldığında yaklaşık %35 arttığı görülmüştür. Bu aradaki %2 lik fark ise, artan hücre geçişi olaylarından kaynaklanmaktadır.

Çok hücreli yöntemlerde bugüne kadar yapılan performans artırma çalışmaları genellikle ilk olayın belirlenmesi aşamasında kullanılan veri yapısı üzerine olmuştur [4, 5, 8]. Fakat bu çalışmaların tüm sistem performansı üzerindeki etkisi çok küçüktür çünkü performans artışı sağlanan kısım, simülasyon için harcanan genel zamanın çok küçük bir kısmını teşkil etmektedir. Bu çalışmada simülasyon hızının arttırımına yönelik daha büyük bir unsur hedeflenmiş ve simülasyonun genelinde çok yüksek sayılabilecek bir performans artışı elde edilmiştir.

Sonuç olarak, tüm sistem performansı düşünüldüğünde, performansı asıl etkileyen unsurun moleküller arası çarpışma hesabı olduğu gerçeğiyle paralel olarak, hücre geçişlerindeki küçük artışa rağmen, toplam çalışma süresinin, yaklaşık hücre sayısının azalma oranıyla doğrusal olarak azaldığı gözlemlenmiştir.

7. Öneriler

Önceki çalışmalarda, yığıt şeklinde gerçekleşmiş öncelik kuyrukları kullanıldığı takdirde simülasyonun performansının daha iyi olduğunu gösterilmiştir. Bu nedenle ileriki çalışmalarda veri yapısı bu yönde değiştirilmesi planlanmaktadır. Buna yanında komşu hücre sayısını daha fazla azaltmak amacıyla farklı hücre tipleri ve farklı hücre dizilimleri denenmeye devam edilecektir. Ayrıca molekül yoğunluğunun ortam simülasyon performansı üzerindeki etkisi araştırılacak ve optimum molekül yoğunluğu belirlenecektir.

8. Bilgilendirme

M. Serdar SORAN'a manevi desteğinden dolayı teşekkürler.

9. Kaynakça

- [1] S. Miller¹ ve S. Luding¹, Event-driven molecular dynamics in parallel, J. of Comp. Phys., Kısım 193, Nüsha 1, (2004)
- [2] Boris D. Lubachevsky, How to Simulate Billiards and Similar Systems, J. of Comp. Phys., Kısım 94, No 2, Sf. 255-283, (1991)
- [3] İlyas KANDEMİR, A Multicell Molecular Dynamics Method, Case Western Reserve University, Source DAI-B 60/03, Sf. 1154, (1999)
- [4] M. Marín, D. Risso ve P. Cordero, Efficient Algorithms for Many-Body Hard Particle Molecular Dynamics, J. of Comp. Phys., Kısım 109, No 2, Sf. 306-317, (1993)
- [5] Rapaport D., The Event Scheduling Problem in Molecular Dynamic Simulation, J. of Comp. Phys., Kısım 34, Sf.184- 201, (1980)
- [6] Masaharu Isobe (Kyushu University), Simple and efficient algorithm for large scale molecular dynamics simulation in hard disk systems, Int. J. Mod. Phys., Kısım 10, 1281, (1999)
- [7] Aleksandar Donev, Salvatore Torquato, Frank H. Stillinger, Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles, J. of Comp. Phys., Kısım 202 Sf. 737-764, (2004)

- [8] Alan T. Krantz , Analysis of an Efficient Algorithm for the Hard-Sphere Problem, ACM Transactions on Modeling and Comp. Simulation, Kısım 6, No 3, (1996)