

T.C.

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ

MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

HÜCRE YAPISININ ÇOK HÜCRELİ OLAY TEMELLİ MOLEKÜL
DİNAMİĞİ SİMÜLASYONLARINDAKİ ETKİSİ

Bilge SORAN

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı

Gebze

2007

T.C.

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ

MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

HÜCRE YAPISININ ÇOK HÜCRELİ OLAY TEMELLİ MOLEKÜL
DİNAMİĞİ SİMÜLASYONLARINDAKİ ETKİSİ

Bilge SORAN

Yüksek Lisans Tezi

Danışmanı Yrd. Doç. Dr. Fatih Erdoğan SEVİLGİN

Bilgisayar Mühendisliği Anabilim Dalı

Gebze

2007

 GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ	YÜKSEK LİSANS/DOKTORA JÜRİ ONAY FORMU
--	---

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ___/___/___ tarih ve ___/___ sayılı kararıyla oluşturulan jüri tarafından ___/___/___ tarihinde tez savunma sınavı yapılan Bilge SORAN ın tez çalışması Bilgisayar Mühendisliği Anabilim Dalı YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE (TEZ DANIŞMANI) : Yrd. Doç. Fatih Erdoğan SEVİLGİN

ÜYE : Yrd. Doç. Yusuf Sinan AKGÜL

ÜYE : Yrd. Doç. İlyas KANDEMİR

ONAY

G.Y.T.E. MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ Yönetim Kurulu'nun ___/___/___ tarih ve ___/___ sayılı kararı.

İMZA/MÜHÜR

ÖZET

TEZİN KONUSU: HÜCRE YAPISININ ÇOK HÜCRELİ OLAY TEMELLİ MOLEKÜL DİNAMİKLERİ SİMÜLASYONLARINDAKİ ETKİSİ

YAZAR: Bilge SORAN

Olay temelli molekül dinamiği simülasyonları, çok parçacıklı sistemlerdeki olayları ardışık olarak ele alarak, sistemin davranışları üzerinde çalışmayı sağlayan bir tekniktir. Bir molekül dinamiği simülasyonunda, gerçekleşecek bir sonraki olay her bir molekül ile diğer bütün moleküller arası muhtemel çarpışmalar hesaplanarak bulunabilir. Ancak, birbirinden uzak iki molekülün çarpışma ihtimali oldukça düşüktür. Çok hücreli yaklaşımda, bir sonraki olayın bulunması için sadece belirli bir alan içindeki moleküllerin incelenmesi yeterlidir. Bu yaklaşımda simülasyon alanı ızgara şeklinde hücrelere bölünmektedir. Bir molekül için kendi içinde bulunduğu hücre ve bu hücrenin komşu hücrelerindeki moleküller incelenerek molekülün karışacağı bir sonraki olay hesaplanabilmektedir. Bu çalışmada, yeni hücre yapıları ve dizilimleri kullanılarak, hücrelerin komşu hücre sayısını 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 %35-40 performans artışı elde edilmiştir.

SUMMARY

THESIS SUBJECT: THE EFFECT OF CELL STRUCTURE IN THE PERFORMANCE OF MULTI-CELL EVENT DRIVEN MOLECULAR DYNAMICS SIMULATION

AUTHOR: Bilge SORAN

Event driven molecular dynamics simulation is an important technique that enables the study on a multi particle system by handling the events sequentially. In a molecular dynamics simulation, at any time, the next collision event that will eventuate can be found by evaluating all possible intermolecular collisions. However, it is so improbable that two molecules that are far away from each other collide. By using multicell approach, to find the next event, it is enough to check the molecules that reside in a defined area. In this approach, the simulation area is divided into cells in a grid manner. So that, a molecules next collision event can be found by looking at the molecules in its owner cell and the molecules at the neighbor cells of the owner cell. In this study, a new cell alignment is proposed to decrease the neighbor count of a cell. Therefore, the search area for a collision event and the simulation time is decreased. At the end of the experiments, when compared with the classical event based molecular dynamics simulation, almost 35-40 % performance gain is obtained.

TEŞEKKÜR

O'na, eşime, anneme ve hocama sonsuz teşekkürler.

İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
1 GİRİŞ	1
1.1 Molekül Dinamiği Simülasyonu Nedir?	3
1.2 Molekül Dinamiği Simülasyonlarında Kullanılan Farklı Yaklaşımlar	3
1.2.1 Doğrudan Simülasyon Monte-Carlo (DSMC)	3
1.2.2 Hesaplamaya Dayalı Molekül Dinamiğinin Simülasyonları (MD)	4
1.3 Molekül Dinamiği Simülasyonlarının Kısa Tarihçesi	5
2 Molekül Dinamiği Simülasyonları	6
2.1 Molekül Dinamiği Simülasyon Metodları	8
2.1.1 Zaman Temelli Simülasyonlar	9
2.1.2 Olay Temelli Simülasyonlar	10
2.1.2.1 Olay Temelli MD Simülasyonlarındaki Sınırlar	11
2.1.2.2 Genel Kabuller	11
2.1.2.3 Olaylar	12
3 Molekül Dinamiği Simülasyonlarında Performans Artırımına Yönelik Çalışmalar	13
3.1 Çok Hücreli Molekül Dinamiği Simülasyonu	14
3.1.1 Olay İşleme Döngüsü	16
3.1.2 Çok Hücreli Olay Temelli MD Simülasyonları Üzerine Yapılan Çalışmalar:	19
4 Çok Hücreli Yaklaşım Getirilen Yenilikler	22

4.1	Uygulanan Hücre Tipleri ve Etkileri	22
5	Kodlama Detayları	25
5.1	Sınıf Şeması	26
5.2	İklendirme	28
5.3	Kullanılan Veri Yapıları	29
5.4	Olay Hesaplama İşlemi	30
5.5	Olay İşleme Döngüsü	36
5.6	Görselleştirme	37
6	Deneyler	38
6.1	Elde Edilen Veriler	39
6.1.1	Veri I	39
6.1.2	Veri II	43
6.1.3	Veri III	47
6.1.4	Veri IV	48
6.1.5	Veri V	50
6.1.6	Veri VI	52
6.2	VerilereDayalı Grafik ve Analizler	53
6.2.1	Simülasyon Zamanı	53
6.2.2	Hücre Geçişİ	54
6.2.3	Duvar Çarpışması	55
6.2.4	Molekül Sayısı-Simülasyon Zamanı İlişkisi	56

6.2.5	Silindir Hücre Tipi İçin Derinlik-Simülasyon Zamanı İlişkisi	57
7	Sonuçlar	58
8	Öneriler	59
9	Özgeçmiş	61

SİMGELER VE KISALTMALAR DİZİNİ

MD	:	Molecular Dynamics - Molekül Dinamiği
DSMC	:	Direct Simulation Monte Carlo
BST	:	Binary Search Tree - İkili Arama Ağacı
LMA	:	Local Minima Algorithm - Yerel Minimum Algoritması
CBT	:	Complete Binary Tree - Tamamlanmış İkili Ağaç
Id	:	Identification Code - Kimlik Bilgisi
Spacing Ratio	:	Moleküller arası mesafenin molekül çapına oranı

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1 Klasik yaklaşımdaki üç boyutlu ve iki boyutlu ortam görüntüsü	8
3.1 Klasik çok hücreli yaklaşımla üç boyutlu ve iki boyutlu ortamın hücrelere bölünmesi – HücreDizilimi1	15
3.2 Çok Hücreli Olay Temelli Molekül Dinamiği Simülasyonunda Olay İşleme Döngüsü	18
4.1 Uygulanan alternatif hücre yerleşimi - HücreTipi2	23
4.2 Uygulanan silindir biçimli hücre şekli - HücreTipi3	24
5.1 Sınıf Şeması	27
5.2 Olay veri yapısı	29
5.3 Olay çeşitleri	30
5.4 Bir olay hesaplama detayları	35
5.5 Olay işleme döngüsü	37
6.1 Spacing Ratio X Simülasyon Zamanı	53
6.2 Spacing Ratio X Hücre Geçişi	54
6.3 Spacing Ratio X Duvar Çarpışması	55
6.4 Molekül Sayısı X Simülasyon Zamanı	56
6.5 Silindir Derinliği X Simülasyon Zamanı	57

TABLOLAR DİZİNİ

<u>Tablo</u>	<u>Sayfa</u>
6.1 Molekül Sayısı 1.000 iken gerçekleşen 10.000 çarpışmanın tanımlı spacing ratiolar ile farklı hücre tiplerinde gerçekleşme zamanları	39
6.2 Molekül Sayısı 8.000 iken gerçekleşen 80.000 çarpışmanın tanımlı spacing ratiolar ile farklı hücre tiplerinde gerçekleşme zamanları	40
6.3 Molekül Sayısı 27.000 iken gerçekleşen 270.000 çarpışmanın tanımlı spacing ratiolar ile farklı hücre tiplerinde gerçekleşme zamanları	41
6.4 Molekül Sayısı 64.000 iken gerçekleşen 640.000 çarpışmanın tanımlı spacing ratiolar ile farklı hücre tiplerinde gerçekleşme zamanları	42
6.5 Molekül Sayısı 1.000 iken gerçekleşen 10.000 çarpışma süresince, farklı hücre tiplerinde yapılan hücre geçişi sayıları	43
6.6 Molekül Sayısı 8.000 iken gerçekleşen 80.000 çarpışma süresince, farklı hücre tiplerinde yapılan hücre geçişi sayıları	44
6.7 Molekül Sayısı 27.000 iken gerçekleşen 270.000 çarpışma süresince, farklı hücre tiplerinde yapılan hücre geçişi sayıları	45
6.8 Molekül Sayısı 64.000 iken gerçekleşen 640.000 çarpışma süresince, farklı hücre tiplerinde yapılan hücre geçişi sayıları	46
6.9 Farklı spacing ratio'larda ve farklı molekül sayılarında gerçekleştirilen duvar çarpışmaları	47
6.10 Spacing ratio 3.5 değerinde simülasyon zamanları	48
6.11 Spacing ratio 4 değerinde simülasyon zamanları	49
6.12 Spacing ratio 4.5 değerinde simülasyon zamanları	49
6.13 Spacing ratio 5 değerinde simülasyon zamanları	50

6.14 Spacing ratio 3.5 deęerinde, silindir hücre tipinde, farklı silindir derinliklerinde simülasyon zamanları	51
6.15 Molekül sayısı 125.000 için simülasyon deęerleri	52

1 GİRİŞ

Molekül dinamiği simülasyonları, çok parçacıklı sistemlerin özellikleri ve davranışı hakkında bilgi veren bir tekniktir[6]. 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 dalayısı 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 çalıştırılabilmesidir[6].

Olay temelli simülasyonlarda ise olası olaylar zaman sırasında işlenir ve moleküller sadece bir olay gerçekleştiğinde güncellenir[8]. 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 elde edilir. Bir molekülün ilk olayının bulunması, diğer bütün moleküller ile çarpışma durumu göz önüne alındığında, N adet molekülün bulunduğu bir sistemde, $O(N)$ 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 azaltılabilir. Bir olay sonrasında sadece olaya katılan moleküllerin olası ilk olayları değişmekte ve yeniden hesaplanması gerekmektedir. Bu durumda, her olayın işlenmesi $O(N)$ zaman almaktadır ki, bu tip simülasyonlarda molekül sayıları çok yüksek olduğu düşünüldüğünde hala oldukça yüksek bir maliyetdir. Bu nedenle simülasyonu etkin bir şekilde çalıştırmak için çeşitli iyileştirme öne sürülmüştür.

Moleküller arası çarpışmanın birbirine yakın moleküller arasında gerçekleşmesi daha olasıdır[1, 2 7]. 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[7]. Bu komşuluğun sınırlarını belirlemek üzere çok hücreli parçalama yöntemi kullanılmıştır[1, 2, 7]. 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. 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 yöntemin kullanılması ile bir olayın işlenme süresinin $O(\log N)$ 'e indirilebileceğini göstermiştir[7]. 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 dinamiği simülasyonlarının performansının nasıl daha fazla arttırılabileceği sorusu üzerinde durulmuştur. 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. Farklı hücre tipleri ve farklı hücre dizilimleri kullanılarak komşu sayısı önce 16'ya sonra 14'e düşürülmüştür. Hücre başına düşen ortalama molekül sayısı değiştirilmeden, komşuluk sayısı azaltıldığı için, simülasyon performansında teorik olarak doğrusal bir artış (yaklaşık olarak 16 komşulukta %37, 14 komşulukta %44) elde edilmesi beklenmektedir. Yapılan deneysel çalışma sonucunda, belli molekül sayılarına kadar, beklenen bu değerlere yakın bir performans artışı elde edilmiştir. Daha yüksek molekül sayıları için artış oranları ve nedenleri sonuçlar bölümünde ele alınmıştır.

1.1 Molekül Dinamiği Simülasyonu Nedir?

Molekül Dinamiği Simülasyonları, kimyasal sistemleri atomik seviye detayında tanımlayan bilgisayar modelleri için kullanılan genel bir terimdir. Bir molekül dinamiği simülasyonunda kimyasal bir sistem, atomların pozisyon ve oryantasyonları yoluyla tanımlanır. Bu simülasyonlar sayesinde Newton'un hareket eşitlikleri uygulanmış bir çok parçacığın davranışı aynı anda incelenerek sistelerin termodinamik ya da kinetik özelliklerini elde edilebilir. Bir gaz sıkıştırıldığında nasıl davranır, sıvılaşması için gerekli basınç nedir gibi sorular teorik olarak cevaplanabilir. Sonuç olarak bilinen mikroskobik modelin makroskobik derivasyonları ile ilgili teorik hipotezler, mikroskobik modeldeki sapmalardan kaynaklanan komplikasyonlarla uğraşmadan incelenebilir ve mikroskobik model ile deneyler teorik yaklaşımlara ihtiyaç duyulmadan direkt karşılaştırılabilir.

1.2 Molekül Dinamiği Simülasyonlarında Kullanılan Farklı Yaklaşımlar

Genel olarak molekül dinamiği simülasyonlarında iki farklı yaklaşım vardır. Bunlardan ilki MD (Molekül Dinamiği - Molecular Dynamics) simülasyonları olarak adlandırılır ve bu yaklaşımda bütün moleküller arası ilişkiler hesaplanır ve bundan dolayı maliyeti çok fazladır. Bu sebeple aynı anda sınırlı sayıda molekülün simülasyonuna olanak sağlar.

Diğer yaklaşım ise DSMC (Doğrudan Simülasyon Monte-Carlo - Direct Simulation Monte-Carlo) simülasyonu olarak adlandırılır ve istatistiksel bilgilere dayanan bir simülasyondur. Bu yaklaşımda moleküllerin rastsal yaklaşımları incelenir. Bu sebeple daha çok molekül içeren simülasyonlar gerçekleştirilebilir. Ancak kullanılan veriler istatistiksel veriler olduğundan tam olarak gerçekçi simülasyon denilemez.

1.2.1 Doğrudan Simülasyon Monte-Carlo (DSMC)

DSMC seyreltilmiş gazların simülasyonunda kullanılan, gaz moleküllerinin istatistiksel davranışlarını inceleyen, hesaba dayalı bir Monte-Carlo algoritması olarak tanımlanabilir. Taneciklerin rastgele hareketi ve istatistiksel teori üzerine oturtulmuştur.

İstatistiklerden elde edilen rastgele verilerin kullanıldığı örnekleme konfigürasyonları denenebilir. Özel örnekleme metodları geliştirilebilir, termodinamik ve yapısal özelliklerle ilgili bilgi sağlar.

DSMC genel olarak tekrarlanan adımlardan oluşur. Her aşamada rastgele seçilmiş bir molekülün hareketi ele alınır. Bir hareketin gerçekleşip gerçekleşmeyeceğine karar vermek için yine rastgele seçilmiş sayılardan faydalanılır. Karar enerji değişiminin uygun olup olmadığına da bakılarak verilir.

1.2.2 Hesaplamaya Dayalı Molekül Dinamiğinin Simülasyonları (MD)

MD ayrık ve anlık etkileşimleri haricinde bağımsız hareket eden, aralarında bir çekim alanı bulunmayan parçacıklardan oluşan sistemlerin simülasyonunda kullanılan etkili bir algoritmadır. Adım-adım bir algoritmayla Newton'un hareket eşitlikleri çözülür[6]. Simülasyon zamanı pikosaniyelerden nanosaniyelere değişir. Ve sonuçta termodinamik, yapısal ve dinamik özelliklerle ilgili bilgi sağlar. Farklı uygulama şekilleri vardır. Simülasyon daha sonra detaylı anlatılacak olan olay temelli ya da zaman temelli olarak yapılabilir.

Molekül Dinamiği simülasyonları paralel ya da seri olarak gerçekleştirilebilir. Parallellleştirilmesinin zorluğundan dolayı olay temelli algoritmalar genelde seri olarak gerçekleştirilir. Dolayısıyla tek işlemci üzerinde yüksek performans sağlamak için verimli algoritmalara ihtiyaç vardır. Zaman temelli algoritmaları paralellştirmek daha kolaydır. Bu sebeple daha yüksek molekül sayılarının aynı anda simülasyonunu görmek daha olasıdır. Paralel algoritmalar bölge ayrıştırma (Domain Decomposition) ve dinamik yük dengeleme (Load Balancing) yöntemine dayanırlar. Seri algoritmalar ise ardışık olay işleme döngüsü üzerine kurulmuşlardır.

Dökümanın bundan sonraki bölümlerinde “hesaplamaya dayalı molekül dinamiği simülasyonu” ibaresi kullanılmayacak kısaca “molekül dinamiği simülasyonu” ibaresi kullanılacaktır.

Sonuç olarak, DSMC ve MD simülasyon teknikleri simülasyonun yapılış amacına göre tercih edilir. Simülasyonun kısa bir sürede, genel bir fikir vermesi hedefleniyorsa,

DSMC tercih edilebilir. Ancak gerçekçi, belli bir zamanda moleküllerin konumu bilgisini sağlayabilecek bir simülasyon isteniyorsa, MD simülasyon tekniği uygulanmalıdır. DSMC ile her olay hesaplanmadığı, dolayısıyla, yapılan hesaplamalar azaltıldığı için çok daha yüksek sayıda molekülün bulunduğu ortamların simülasyonunu gerçeklemek daha kolaydır. Oysa MD simülasyonlarında işlemci kısıtı ön plandadır. Bu nedenle MD simülasyonları ile ilgili yapılan çalışmalar genelde performans arttırımına yöneliktir. Bu kısıtına rağmen, gerçeğe en yakın sonuçlar ancak MD simülasyonları ile sağlanabilir.

1.3 Molekül Dinamiği Simülasyonlarının Kısa Tarihçesi

Molekül dinamiği simülasyonları üzerindeki ilk çalışmalar 1957 yılında Alder ve Wainwright tarafından yapılmıştır. Simülasyonları katı-sıvı geçiş fazında, katı disk olarak modellenmiş birçok molekülle yapmışlardır. Sonuç olarak dışarıdan herhangi bir kuvvet uygulanmamasına rağmen sistemin kristalize olduğunu görmüşlerdir. Bu çalışmalar bugünlerin simülasyon çalışmalarını önemli ölçüde etkilemiştir. Katı disk sisteminde, molekül dinamiği sadece çarpışmalardan ve düz bir çizgi üzerinde ilerlemeden ibaret olarak kabul edilmiştir[3].

Simülasyonlar, sabit zaman aralıklarında sisteme Newton'un hareket eşitlikleri uygulanmak suretiyle çalıştırılabilceği gibi(Zaman Temelli Molekül Dinamiği), sistemde bir olay gerçekleştiğinde sistemi güncellemek şeklinde de(Zaman Temelli Molekül Dinamiği) çalıştırılabilir. Molekül çiftleri üzerinden direk hesaplamalar N sayıda molekül için $O(N^2)$ kompleksliğe sahiptir. M.Isobe'a göre ise katı disk sisteminde, kullanılacak veri yapısının ve gelecek olayların kuyruğunun ne kadar iyi işlendiği sistemin performansını belirleyici unsurdur[3]. Bu konuda Rapaport bir iyileştirme önermiştir[7]. Bu öneri çok hücreli metoda dayanmaktadır ve detayları [Molekül Dinamiği Simülasyonlarında Performans Arttırımına Yönelik Çalışmalar](#) bölümünde yer almaktadır.

2 MOLEKÜL DİNAMİĞİ SİMÜLASYONLARI

Molekül dinamiği simülasyonu, bir moleküler sistemdeki moleküllerin mikroskopik davranışlarından faydalananarak makroskopik davranışlarını anlamaya yönelik bir tekniktir. Bu teknikte moleküller bir kontrollü ortam içinde konuşlandırılmışlardır. Gerçeğe en yakın sonuçları elde edebilmek için, ortamdaki molekül sayısının laboratuvar ortamında elde edebilecek molekül sayısına olabildiğince yakın olması gerekmektedir. Dolayısıyla sistemin performansı aynı anda simülasyona katılan molekül sayısı ile yakından ilgilidir.

MD simülasyonu, özellikleri önceden tanımlı, kontrollü bir ortamdaki her bir molekülün ayrı ayrı modellenmesinden oluşmaktadır. Bu güne kadar farklı özelliklerde ortamlar tanımlanmış, farklı koşullarda simülasyonlar gerçekleştirilmiştir. Yapılan bu çalışmalarda simülasyonun geneline etki edebilecek, simülasyon performansını arttırmaya yönelik bir takım varsayımlar yapılmıştır. Molekül simülasyonlarında tanımlanan kontrollü ortamlar genellikle yapılan çalışmalarda şekilsel bakımdan dikkörtgenler prizması şeklinde kapalı bir kutu olarak tanımlanmıştır. Ortamın diğer nitelikleri simülasyonun tipine ve hedefine göre değişiklik göstermektedir. Simülasyonların tipiyle kasıt, simülasyonun gerçek zamanlı ya da önceden tanımlı koşullarda gerçekleşmesidir. Gerçek zamanlı simülasyonlarda, ortam simülasyonun çalışması esnasında değiştirilebilir ve bu değişikliklerin eş zamanlı olarak moleküller üzerindeki etkisi izlenebilir. Örneğin, bir piston vasıtasıyla ortam basıncı kontrol edilebilir ya da ısıtma yoluyla ortam sıcaklığı değiştirilebilir. Bunların aksine, gerçek zamanlı olmayan simülasyonlarda ortam koşulları simülasyon başlamadan önce girdi olarak verilir. Örneğin, belirli basınç ya da sıcaklık koşullarında moleküllerin davranışları izlenip yorumlanabilir.

Her simülasyon ortamının bir toplam enerjisi vardır. Gerçek zamanlı sistemlerde ortamın toplam enerjisi dış etkenlerle değişir ve simülasyon boyunca bu enerji değişiminin kontrol edilmesi gerekir. Bu nedenle koşulları önceden tanımlı ortamlar, gerçek zamanlı olarak özellik değiştirebilen ortamlara göre hesaplama maliyeti açısından daha avantajlıdır. Son yıllarda yapılan çalışmalarda, ortamı ısıtmak, basıncı arttırmak ya da farklı uygulamalarla ortam enerjisisini değiştirmek gibi faktörlerle, molekül dinamiği

simülasyonları gerçekleştirilmiş olmasına rağmen, genel olarak yapılan çalışmalarda, simülasyon boyunca, toplam enerjinin korunduğu varsayılmıştır.

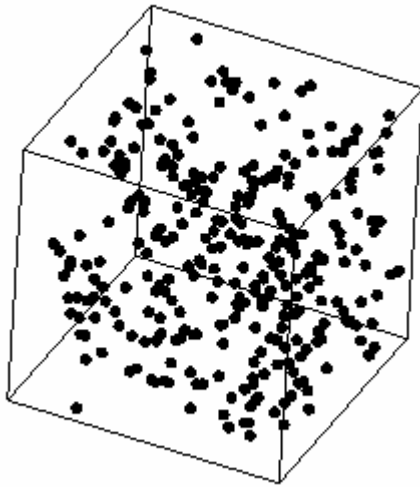
Bir moleküler sistemde her molekül tek atomlu yapıda olmak zorunda değildir, farklı şekillerde farklı elementlerin kombinasyonu formatında modellenebilir. Ancak, hesaplamaların kolaylaştırılması amacıyla, genellikle yapılan çalışmalarda moleküllerin küresel oldukları kabul edilir. Bu modele katı küre modeli denilmektedir.

Doğada her molekülün hem doğrusal hızı hem de açısal hızı ve bunların sonucunda bir enerjisi mevcuttur. Zaten ortamın enerjisi de, bir bakıma, ortamdaki moleküllerin enerji toplamına eşittir. Her molekülün kendine ait bir doğrusal hızı olduğundan moleküller arası çarpışma da kaçınılmazdır. Çarpışmalar ise farklı şekillerde modellenebilir. Çarpışma sonunda moleküller arası tepkimeler gerçekleşebilir. Simülasyon teknikleri bu tip tepkimelerin izlenmesi amacıyla da kullanılabilir ancak bu tip simülasyonların maliyeti çok yüksektir ve bu sebeple simülasyon ortamına aynı anda dahil edilebilen molekül sayısı kısıtlıdır. Tepkimeler dışında, oluşan çarpışmalar esnek ya da esnek olmayan olarak gerçekleşebilir. Esnek olmayan çarpışmalar modelleme ve hesaplama yönüyle çok daha karmaşıktır. Simülasyondaki asıl amaç moleküllerin şekilsel özelliklerinden ziyade, yüksek sayıda molekül ile sistemin genel davranışını takip etmektir. Bu nedenle bugüne kadar yapılan çalışmalar genelde çarpışmaları esnek olarak varsaymışlardır.

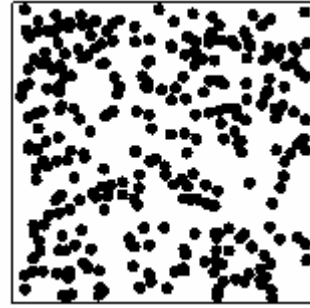
Sonuç olarak molekül dinamiği simülasyonu çalışmalarındaki belli başlı kabuller, ortamdaki-moleküle şöyle sıralanabilir:

- harici sınırlar
- ortam enerjisinin korunumu,
- moleküller arası esnek çarpışmalar,
- küresel, atomik, simülasyon boyunca esnek çarpışmalarla ya da diğer çevresel faktörlerle değişebilen doğrusal bir ilk hızı olan molekül modeli.

Bu kabuller ışığında ortam kapalı, dışarıdan bir etki altında bırakılmayan yani sabit enerjili, dikdörtgenler prizması şeklindedir. Bu da moleküllerin etkileşebileceği, ortama ait 6 tane duvar yüzeyinin olduğu anlamına gelir. Bu modelin görüntüsü Şekil2.1’de iki ve üç boyutlu olarak gösterilmiştir. Simülasyonun kabullerine bağlı olarak değişmekle beraber, her molekülün, koordinat eksenleri boyunca uzanan bileşenleri olan, molekülün hareket yönünü belirleyen bir doğrusal hızı, çarpışma sonrası yöneliminde belirleyici rol oynayan bir açısal hızı, kütlesi, hacmi ve ağırlık merkezi vardır.



Ortamin 3 boyutlu görüntüsü



Ortamin bir yüzey görüntüsü

Şekil 2.1 Klasik yaklaşımdaki üç boyutlu ve iki boyutlu ortam görüntüsü

2.1 Molekül Dinamiği Simülasyon Metodları

Molekül dinamiği simülasyonlarında kullanılan, simülasyonun amacına yönelik, farklı teknikler vardır. Simülasyonlar çok işlemci üzerinde paralel olarak gerçekleştirilebileceği gibi tek işlemci üzerinde de gerçekleştirilebilir. Genel olarak molekül dinamiğinin simülasyonlarında kullanılan algoritmalarda olay temelli ve zaman temelli

olmak üzere iki farklı yaklaşım vardır ve bu yaklaşımlar paralelleştirme açısından farklı niteliktedirler[6]. Zaman temelli algoritmalar global bir zaman değerine sahiptir ve her zaman değişikliği için tüm moleküllerin durumunu güncellenir. Dolayısıyla çok sayıda molekül olduğu düşünüldüğünde maliyeti çok yüksektir. Olay temelli algoritmalar ise olayları belirler ve sıraya koyar. Molekülün durumu sadece kendisini ilgilendiren bir olay gerçekleştiği zaman güncellenir. Dolayısıyla ortamın ve moleküllerin özelliklerine bağlı olarak değişmekle birlikte genel olarak zaman temelli algortimalara göre tek işlemci üzerinde masrafı daha azdır. Sonuç olarak, hangi tekniğin uygulanacağı simülasyonun amacına göre belirlenmelidir.

2.1.1 Zaman Temelli Simülasyonlar

Bu tip simülasyonlarda kullanılan metod sınırlı sabit bir zaman dilimi üzerine ve parçacıkların bu zaman dilimleri içinde eşitlikleri sağlaması üzerine kuruludur. Yani zamanla birlikte moleküllerin hareketleri benzetilir. Belli bir zaman anında moleküllerin durumu izlenebilir.

Zaman temelli algoritmalar genel olarak N-cisim gibi çok parçacıklı sistemlerin simülasyonunda kullanılan bir yaklaşımdır. Molekül dinamiğinin simülasyonunda da, dolayısıyla, bu tip bir tekniğin uygulanması oldukça uygundur. Zaman temelli algoritmalarda, simülasyon akışı belirli zaman aralıklarına bölünmüştür. Simülasyon her bir zaman periyodunun sonunda güncellenir[4]. Bir parçacığa ait özelliklerin güncellenmesi, ortamla ilgili özelliklerin güncellenmesi bu güncellemelerin başlıcalarıdır. Bu yaklaşımdaki temel problem zaman aralıklarının iyi belirlenmesidir. Bir periyod aralığında herhangi bir molekül en fazla bir olaya katılmış olmalıdır. Eğer birden fazla olaya katılması söz konusu ise bu olaylardan bir kısmı kaçırılacaktır. Eğer bir zaman aralığında hiç bir molekül bir olaya tabi olmuyorsa bu da gereksiz güncelleme demektir. Ayrıca, her zaman aralığında molekülleri güncellemek, hata miktarının artmasına da neden olur. Bu da neticede şu şekilde özetlenir: Eğer zaman aralıkları çok kısa olursa, gereksiz yere simülasyon güncellemelere tabi tutulacaktır. Eğer zaman aralıkları çok uzun olursa, arada gerçekleşmesi muhtemel bazı olaylar kaçırılacaktır. Dolayısıyla simülasyonun akışı bozulacaktır. Uygun zaman aralıklarını belirlemek için olay

periyodları istatistiksel olarak değerlendirilmelidir. Ya da daha gelişmiş uygulamalarda, simülasyonun akışına göre değişken zaman periyodları kullanılabilir.

Bunlara ilaveten her bir zaman periyodunun sonunda tüm ortamın güncellenmesi oldukça maliyetli bir iştir. Ancak, her bir molekülün simülasyon akışıyla paralel olarak güncel kalması için bu yapılmak zorundadır. Bu tip uygulamalarda, moleküllerin güncellenmesi asıl maliyet unsurunu oluşturmaktadır. Dolayısıyla maliyet analizi için bu verinin değerlendirmesi bize yeterli bilgiyi verecektir. Örneğin, simülasyon T zaman periyoduna bölünmüş ve ortamda N sayıda molekül mevcut ise, simülasyon boyunca, $O(T*N)$ tane güncelleme yapılmaktadır ki aslında bu moleküllerin bir çoğu sadece bir kaç kez bir olaya tabi olmalarına rağmen T kez güncellenmektedirler. Ancak zaman temelli simülasyonlar tanımlarındaki paralellikten dolayı, olay temelli simülasyonlara göre çok daha etkili bir şekilde çok işlemci üzerinde dağıtık bir şekilde çalıştırılabilir. Bu da performans açısından olay temelli simülasyonların önüne geçebilmeleri anlamına gelmektedir. Ayrıca zaman temelli simülasyonların bir avantajı da, her an tüm moleküllerin olması gereken yerde olması sebebi ile, elde edilecek makroskopik verilerin her an doğru olmasıdır. Olay temelli simülasyonlarda ise daha çok simülasyonun son durumuna bakılır. Bu nedenle zaman içinde sistemin nasıl değiştiğinin görüntülenmek istendiği durumlar için zaman temelli simülasyonlar daha uygundur.

2.1.2 Olay Temelli Simülasyonlar

Olay temelli algoritmalar da, genel olarak çok parçacıklı sistemlerin simülasyonunda kullanılan bir yaklaşımdır. Bu yaklaşımda tüm simülasyon olaylar üzerine kuruludur. Bir olay gerçekleşmesi başka olayların oluşmasına sebebiyet verebilir. Simülasyonun ilerleyişi olayların zamanının gelmesi ve gerçekleşmesi şeklinde olur. Yani bir olayın gerçekleşmesinden gerçekleşmesine zaman ilerler. Aradaki zamanlarda moleküllerin durumuyla ilgilenilmez ve bir moleküle ait özellikler, sadece bir olay gerçekleştiğinde güncellenir.

Simülasyonun asıl akış elementleri olaylar olduğundan, bu yaklaşımda performans olarak oldukça iyi sonuçlar elde edilebilir. Simülasyon olayların takibinden ibarettir. Her olayın işlenişinden sonra, meydana gelebilecek bir sonraki olay hesaplanır ve daha

sonraki adımlarda da bu olay işlenir. Simülasyon boyunca ardışık bir şekilde olayların işlenmesi devam eder. Ancak, olayların ardışık yapısından dolayı, olay temelli simülasyonların çok işlemci üzerinde dağıtık ve paralel olarak çalıştırılması oldukça zordur[6]. Bu yüzden paralelleştirme çalışmaları olmasına rağmen, yapılan çalışmalar genelde tek işlemci üzerinde iyileştirmelere yöneliktir.

2.1.2.1 Olay Temelli MD Simülasyonlarındaki Sınırlar

MD simülasyonlarında moleküllerin tüm etkileşimleri incelendiğinden, genel olarak büyük hafızalara ve yüksek işlemci hızlarına ihtiyaç duyulmaktadır. Dolayısıyla çok sayıda molekülün simülasyonunu yapabilmek için karşılaşılabilecek başlıca limitler hafıza ve hız olarak sıralanabilir.

2.1.2.2 Genel Kabuller

Olay temelli bir molekül dinamiği simülasyonunda yapılan kabuller genel molekül dinamiği simülasyonlarındakilerle aynıdır. Kapalı bir kutu içinde belli ilk hızları olan moleküllerin etkileşimleri incelenir. Simülasyonun başından sonuna kadar molekül sayısının sabit kaldığı düşünülmektedir. Ayrıca bu dikdörtgenler prizması şeklindeki kutunun duvarlarında belli sınır özellikleri olduğu varsayılabilir. Moleküller kutunun duvarları ile etkileşirken bu sınır özelliklerine uygun hareket edeceklerdir.

Moleküllerin tek atomdan oluştukları, katı küre şeklinde oldukları ve fizik kurallarına uygun hareket ettikleri kabul edilmektedir. Ayrıca moleküllerin birbirlerini itip çekmedikleri ve dolayısıyla açısal hızlarının olmadığı, sadece doğrusal hızlarının olduğu ve doğrusal bir yörüngede hareket ettikleri varsayılmaktadır.

Tüm moleküller arası çarpışmaların elastik olduğu ve anlık çarpışmaların sadece ve sadece iki molekül arasında yapıldığı kabul edilmektedir. Daha önce yapılmış çalışmalarda üç molekülün aynı anda çarpışmasının simülasyonu ile iki molekülün aynı anda ikişer ikişer çarpışmalarının simülasyonunun aynı sonuçları verdiği ortaya konulmuştur.

2.1.2.3 Olaylar

Olay, simülasyonun belli anki durumunu etkileyecek, simülasyondaki bir ya da birden fazla parçacığın hareketinden kaynaklanan herhangi bir etmendir. Sınırlı bir ortamda gerçekleşen molekül dinamiği simülasyonlarında, moleküllerin hareketlerinden dolayı meydana gelen, temelde iki tip olay vardır. Bunlar; moleküller arası çarpışmalar ve bir molekülün simülasyon ortamının bir duvarına çarpmasıdır. Bir olay çeşitli özellikler taşır. Bunlar, olayın zamanı, olayın tipi, olaya karışan moleküllerdir. Olayın zamanı, olayın gerçekleşme anını ifade eder. Olay zamanları başka olaylara bağıntılı olarak düşünülmez. Genel zaman akışına göre belirlenir. Her olay belli molekül ya da moleküller tarafından gerçekleşir. Olaydan etkilenen moleküller olaya karışan moleküller olarak belirlenir.

Tip1: Moleküller Arası Çarpışma

Her molekülün belli biri hızı olduğundan kapalı bir ortamda bulunan moleküllerin çarpışması kaçınılmazdır. Bu sebeple gerçekleşecek olaylar hesaplanırken her molekülün diğer tüm moleküllerle çarpışıp çarpışmayacağı kontrol edilmelidir. Ortamdaki molekül sayısı N kabul edilirse, bu bir molekül için, karmaşıklığı $O(N)$ olan bir işlemdir.

Tip2: Molekülün Duvara Çarpması

Kapalı bir ortamda bulunan moleküllerin ortamın duvarlarına çarpması da kaçınılmazdır. Ortamdaki her molekülün olay hesaplama periyodunda bu duvarlara çarpıp çarpmayacağı kontrol edilmelidir. Ortamın şekil kısıtı olmadığı düşünülürse; duvar sayısının `WALL_COUNT` olduğu durumda, bu hesaplama bir molekül için $O(WALL_COUNT)$ karmaşıklığında ve `WALL_COUNT` değerinin sabit olduğu varsayıldığında $O(1)$ maliyetindedir.

3 MOLEKÜL DİNAMİĞİ SİMÜLASYONLARINDA PERFORMANS ARTIRIMINA YÖNELİK ÇALIŞMALAR

Bu çalışmaların en başında simülasyonu çok işlemci üzerinde çalıştırmak gelir. Daha önce de değinildiği gibi olay temelli algoritmalarda simülasyonun asıl akış elementleri olaylar olduğundan ve olayların ardışık yapısından dolayı, çok işlemci üzerinde dağıtık ve paralel olarak çalıştırılması oldukça zordur. Bu yüzden paralelleştirme çalışmaları olmasına rağmen[6], yapılan çalışmalar genelde tek işlemci üzerinde iyileştirmelere yöneliktir. Bu iyileştirmelerin başlıcası ise detayları aşağıda verilecek olan çok hücreli yaklaşımdır[3].

Olay temelli simülasyonlarda aynı anda simülasyona dahil olması beklenen ortamdaki molekül sayısı milyonlar seviyesindedir. Buna ilaveten simülasyonlardaki ana maliyet unsuru daha önce de belirtildiği üzere olay hesaplamaları sırasında moleküller arası çarpışmaların kontrol edilmesidir. Bu nedenle, eğer bu noktada bir iyileştirme sağlanabilirse, sistemin geneli üzerinde önemli bir performans artışı sağlanabilir.

Çok hücreli yaklaşımdan genel anlamda anlaşılması gereken ortamın daha küçük alt parçalara bölünmesidir. Bu bölünme, farklı kriterlere göre, farklı amaçlar için, farklı şekillerde yapılabilir. Bir ortamın alt parçalara ayrıştırıldığı durumdaki her bir alt parçaya “hücre” adı verilir. Bu hücrelere ayrıştırma sayesinde, bir molekülle ilgili bir çarpışma olayı hesaplanılacağında ortamdaki tüm moleküllere değil yalnızca, o molekülün içinde bulunduğu hücredeki moleküllere ve bu hücreyi çevreleyen hücrelerdeki moleküllere bakılır. Moleküllerin yarı çapları olduğundan ve bulundukları hücreler merkezlerine göre belirlendiğinden, merkezleri komşu iki hücrede olan moleküller bu hücre sınırlarında çarpışabilirler. Bu nedenle yalnızca molekülün içinde bulunduğu hücrenin sahip olduğu moleküllerle kontrol etmek yeterli değildir. Komşu hücrelerdeki moleküllere de bakılmalıdır. Bu yaklaşım sayesinde her molekül için araştırma alanı milyonlar seviyesinden, çok daha küçük rakamlar seviyesine daraltılmış olur. Takip eden bölümlerde bu yaklaşımın en genel halinin, olay temelli algoritmalarla birlikte kullanıldığı durumlar ele alınacak, elde edilen avantaj ve dezavantajlar anlatılacaktır.

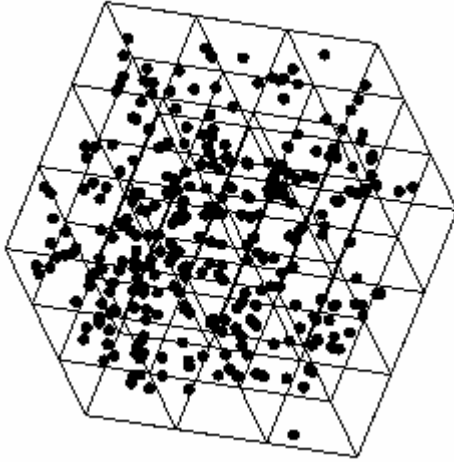
3.1 Çok Hücreli Molekül Dinamiği Simülasyonu

Çok hücreli yaklaşım, molekül dinamiği simülasyonlarını hızlandırmaya yönelik uygulanan, çok genel bir yöntemdir. Genel olarak kapalı bir kutu içindeki moleküller düşünüldüğünde ve moleküllerin bağımsız hareket edebilecekleri yolun kısa olduğu düşünüldüğünde; uzaktaki moleküllerin, birbirleriyle çarpışma olasılıkları oldukça düşüktür. Bu sebeple kutu hücelere bölünmüştür ve belli komşuluktaki hücelerin içindeki moleküllerin birbirleriyle çarpışma ihtimallerinin daha yüksek olduğu düşünülmüştür. Bu yolla hedeflenen, ortamı daha küçük parçalara bölerek birbirleriyle etkileşme ihtimali düşük olan moleküllerin etkileşmelerini hesaplama maliyetinden kurtulmaktır. Bu durum herhangi bir çarpışmanın kaçırılmasına neden olmamaktadır, çünkü moleküllerin bir hücreden bir hücreye geçmesi de bir olay olarak tanımlanmış ve ele alınmıştır. Bir molekülün ilk olayı uzaktaki bir molekülle çarpışmak olsa bile, molekül gerekli hücre geçişlerini yapar ve çarpışacağı moleküle ulaşır. Ancak ileriki bölümlerde açıklanacağı üzere, böyle bir uygulamanın performansı arttırması için ortamdaki molekül yoğunluğunun yüksek olması gerekir. Aksi halde beklenen performans artışından söz edilemez.

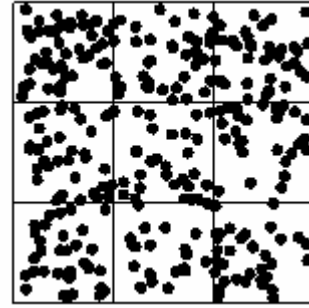
Birbirine uzak moleküllerin etkileşme ihtimali, birbirlerine yakın moleküllere göre çok daha düşük olduğundan, ortamı birbiriyle etkileşme ihtimali yüksek olan moleküllerin bir arada olacağı şekilde parçalara bölmek hesaplama miktarını azaltmak için en iyi yöntemdir. Bu sayede hesaplama miktarı azalacak ve simülasyon genelinde büyük kazanımlar sağlanacaktır. Burada asıl problem ise ortamın kaç parçaya ne şekilde bölüneceğidir. Her bir hücre ortalama kaç tane molekül içermelidir? Hücrelerin geometrik şekli ne olmalıdır? Hücreler arası etkileşim ne şekilde kontrol edilmelidir?

Ortamı alt parçalara yani hücelere ayırmanın en kolay yolu dikdörtgenler prizmasını daha küçük dikdörtgenler prizmalarına bölmektir. Bu sayede, her bir alt prizmada birbirine yakın moleküller yani etkileşme ihtimali yüksek moleküller kalacak ve problem daha küçük alt problemlere dönüştürülmüş olacaktır. Daha önceden yapılmış çalışmalarda simülasyon bir hücrede bulunması gereken optimum molekül sayısı 8 olarak

belirlenmiştir. Yani ortamdaki molekül sayısı N ise hücre sayısı $N/8$ olmalıdır. Ortamın bu şekilde modellenmiş hali Şekil 3.1 de görülebilir.



Ortamın 3 boyutlu görüntüsü



Ortamın bir yüzey görüntüsü

Şekil 3.1 Klasik çok hücreli yaklaşımla üç boyutlu ve iki boyutlu ortamın hücelere bölünmesi – HücreDizilimi1

Şekil 3.1'deki gibi yapılmış bir paylaşımda, farklı hücrelerdeki moleküllerin de birbirleriyle etkileşimi olma ihtimali söz konusudur. Çünkü moleküllerin bulundukları hücreler merkezlerinin yerine göre belirlenir. Merkezleri farklı hücrelerin içinde olan moleküller, içinde bulundukları hücreler birbiriyle bir sınıra sahipse, yarıçaplarından dolayı, bu sınırdan geçebilirler. Bu şekilde oluşabilecek olayları da ele alabilmek için bir “komşuluk” tanımı yapılır. Her hücre, yüzeyinin herhangi bir noktasının değdiği tüm diğer hücrelerle komşudur. Bu durumda yukarıda bahsedildiği şekilde ortam hücelere ayrıldığında her bir hücrenin 26 tane komşu hücresi olur. Bir molekülün diğer moleküllerle çarpışma olayının varlığını kestirebilmek için, o molekülün içinde bulunduğu hücredeki diğer moleküller ve o hücrenin komşu hücrelerinde bulunan tüm moleküllerle etkileşimi kontrol edilmelidir. Çünkü birbirine komşu iki farklı hücrede bulunan moleküller, yukarıda açıklandığı gibi, bu hücrelerin birbirleriyle olan sınırlarında çarpışabilirler.

Çok hücreli yaklaşımın kullandığı durumda, moleküllerin yer değiştirmelerinden kaynaklanan, hücre değişimi gereklidir. Moleküller ait oldukları hücreden komşu bir hücreye özgürce geçebilmelidirler. Dolayısıyla, asıl olay tanımlarına, hücre geçişini kontrol etmek amacıyla bir yenisi daha eklenmelidir. Bu durumda başta yapılan kabullerden devamla, bir hücrenin 26 tane komşusu olduğu durumda, bir molekülün yapması muhtemel olaylar hesaplanırken komşu 26 hücreye geçme durumları da kontrol edilmelidir.

Özetle, bir molekül dinamiği simülasyonu, olay temelli olarak bir algoritma ile çok hücreli bir yaklaşımda gerçekleştirildiğinde, bir molekül için kontrol edilmesi gereken olay tipi artsa da, yüksek molekül sayılı ortamların simülasyonunda, olay hesaplamaları sırasında en önemli etmen olan moleküller arası çarpışmanın maliyeti azalmıştır. Çünkü, her defasında bir molekül için ortamdaki tüm diğer moleküllerle çarpışma olasılığının hesaplanması yerine, çarpışma olasılığı yüksek olan civar pozisyonlardaki moleküllerle olan etkileşimi kontrol edilmiştir. Bu sayede elde edilen sistem performansındaki artış daha önceki çalışmalarla ispatlanmıştır. Ancak, eğer ortam seyreltikse, yani ortamdaki molekül sayısı belli bir orandan az ise, bu yöntem daha maliyetli olabilir. Çünkü moleküller çoğunlukla, bir molekülün duvara çarpması ya da moleküller arası çarpışma gibi gerçek bir olay işlenene kadar, sanal bir olay olarak da nitelenebilecek, bir çok hücre geçişi yapmak zorunda kalacaklardır. Bu da sistemin toplam performansını düşürecektir.

Sonuç olarak, şartlar uygun seçildiği takdirde, hücrelere ayırma yöntemi sayesinde bir molekülün katılacağı en yakın olayın hesaplanması için gereken işlem $O(N)$ den $O(c)$ 'ye düşmektedir. Buradaki c değeri, bir hücre ve komşularındaki ortalama molekül sayısınıdır ve sabit bir sayı olması sağlanmaktadır. Bu kazanç sağlanırken moleküllerin bir hücreden diğerine serbestçe geçmesi sağlandığından herhangi bir olay da atlanmaz. Çünkü, birbirine uzak iki molekülün çarpışması söz konusu ise, o moleküller belli hücre geçişlerini yapıp, tanımlı bir yakınlığa ulaştıklarında söz konusu olay hesaba katılacaktır.

3.1.1 Olay İşleme Döngüsü

Bir olay temelli çok hücreli molekül dinamiği simülasyonunda temel olarak 3 tip olay vardır. Bu olaylar bir molekülün ortam duvarlarından birine çarpması, moleküller

arası çarpışma, bir molekülün bir hücreden diğerine geçmesidir. Bundan sonraki bölümlerde sırasıyla bu olaylar WALL_COLLISION, INTERMOLECULAR_COLLISION, CELL_PASSING olarak anılacaktır.

Çok hücreli bir molekül dinamiği simülasyonunda bir olayın [Bölüm2.2.2.3](#)'de belirtilenlerle birlikte nitelikleri; olayın zamanı, olayın tipi, eğer olay moleküller arası çarpışma ise, çarpışılacak molekül bilgisi, hücre geçişi ise geçilecek hücre bilgisi, olay duvar çarpışması ise, çarpışılacak duvar bilgisi olarak sıralanır.

Bir molekülün gerçekleyeceği ilk olay hesaplanırken, komşu hücelere geçebileceği en yakın zaman, eğer molekül ortamın sınır hücelerinden birinde bulunuyorsa bulunduğu hücrenin bir duvarına çarpma zamanı, ve bulunduğu hücre ve komşu hücelerdeki moleküllerle yapabileceği muhtemel molekül çarpışmaları ve muhtemel hücre geçiş zamanları hesaplanmalıdır. Daha sonra bu muhtemel molekül çarpışma zamanlarının ilki belirlenmelidir. Daha sonra, elde edilen bu bilgiler ışığında, molekülün gerçekleştireceği ilk olayın zamanı ve tipi belirlenmelidir. Bu olayların minimum zamalı olayı molekülün gerçekleştireceği ilk olaydır.

Simülasyonun geneli olay temelli olduğundan her adımda bir olay işlenerek ilerlenir. Dolayısıyla simülasyonun genelinde gerçekleşecek ilk olayın da bulunması gerekir. Buna da ancak ortamdaki hücelerin ilk olaylarının karşılaştırılmasıyla karar verilebilir. Her hücrenin ilk olayı hesaplandıktan sonra bulunan olayların en küçük zamanlısı simülasyonun da ilk olayıdır. Benzer şekilde simülasyon akışı sırasında, şimdiki zamandan sonra gerçekleşecek ilk olay da hücelerin ilk olaylarının minimum zamanlısının seçimiyle bulunur.

Simülasyon sırasında gerçekleşecek olaya ve bu olayın ilgili olduğu moleküle ve dolayısıyla hücreye karar verildikten sonra, olay, çeşidine göre işlenir. Daha sonra varsa olaya katılmış diğer hüceler ve diğer moleküllerle birlikte asıl olayın sahibi olan molekül ve hücre için gelecek olaylar hesaplanır. Böylece o andan sonraki simülasyon akışı sırasında tekrar en küçük olaya karar verilebilir. Bu şekilde simülasyon boyunca olay işleme döngüsü devam eder. Olay işleme döngüsünden çıkmanın iki yolu vardır. Bunlar

ortamda işlenecek hiç olay kalmaması ya da simülasyon zamanının bitmesidir. Olay işleme döngüsünün kodumsu şekli Şekil 3.2 de gösterilmiştir.

1. Her molekülün ilk olayını hesapla.
2. Her hücrenin ilk olayını hesapla.
3. Simülasyonun ilk olayını hesapla.
4. Simülasyonun ilk olayını gerçekleştir.
5. Olaya karışan moleküllerin ilk gelecek olaylarını güncelle.
6. Olaya karışan hücrelerin ilk gelecek olaylarını güncelle.
7. Adım 3'e geri dön.

Şekil 3.2 Çok Hücreli Olay Temelli Molekül Dinamiği Simülasyonunda Olay İşleme Döngüsü

Şekil 3.2'deki hesaplamaların maliyeti ise; ilk olayların hesaplanması ($O(N^2)$), her aşamada bir sonraki olayın belirlenmesi, her aşamada bir olayın işlenmesi maliyetlerinin toplamına eşittir. Her aşamada bir sonraki olayın belirlenmesi ise, bir önceki olaya karışan moleküllere ait gelecek olayların hesaplanması ve tüm olaylar arasından en küçüğünü belirleme işlerinden oluşur. Bir önceki olaya karışan moleküllere ait gelecek olayların hesaplanması ise, çok hücreli yaklaşımda, $O(c)$ dir. Tüm olaylar arasından en küçüğünü belirlemenin maliyeti ise olayların saklandığı veri yapısına göre değişir. Sıralı düz bir liste tutulduğunda maliyet $O(N)$ olabilirken, heap ya da BST gibi bir veri yapısı kullanıldığında maliyet $O(\log N)$ olur. Her aşamada bir olayın işlenmesinin maliyeti ise sadece ilk olayı alıp yerine getirmek olduğundan, $O(1)$ dir. Dolayısıyla çok hücreli yaklaşımda simülasyonun her aşamadaki toplam maliyetini belirleyen unsur, o aşamadaki minimum zamana sahip olan olayı belirlemede kullanılan veri yapısı gibi görünmektedir. Ancak bu çalışma ile c sabitinin simülasyon hızı üzerindeki etkisinin, bir aşamada minimum zamana sahip olan olayı belirlemede kullanılan veri yapısından çok daha büyük olduğu gösterilmiştir.

3.1.2 Çok Hücreli Olay Temelli MD Simülasyonları Üzerine Yapılan Çalışmalar:

Çok hücreli yaklaşım molekül dinamiği simülasyonlarının hızlandırılması açısından bugüne kadar gerçekleştirilmiş en önemli yöntemdir. Daha sonraki aşamalarda ise bu yöntemin hızlandırılmasına yönelik çalışmalar yapılmıştır. Bu konuda bugüne kadar yapılan çalışmalarda ise, hızlanmanın en etkin olarak, bir sonraki en yakın minimum olayı belirlemede kullanılan veri yapısının farklı yapıda olmasıyla sağlanabileceği düşünülmüştür. Bu konuyla ilgili önemli çalışmalar 90lı yıllarda Rapaport tarafından yapılmıştır[3].

Rapaport'un çalışmaları[7]:

- Algoritma çok hücreli yonteme dayanır.
- Çarpışma olayı ve hücre geçişi olayları olay zamanını tutan listelerde saklanır.
- En küçük olay zamanı ikili arama ağacı (BST) kullanılarak aranır.
- Bu olay olduğunda, molekül çifti ya da molekül-hücre çifti (olay tipine göre) ikili arama ağacından silinir ve ilgili molekül için olay zamanı tekrar hesaplanır ve ikili arama ağacı güncellenir.
- BST yeni nodeları ekleyerek tekrar oluşturulur.
- Aynı işlemler tekrar edilir.

Rapaportun algoritmasının bir olay hesabi için ortalama kompleksliği; minimum zamanlı olaya binary search tree kullanılarak karar verildiği için ortalama $O(\log N)$ seviyesindedir. $O(N)$ den $O(\log N)$ 'e düşüş sayesinde hesaplama zamanından büyük bir kazanç elde edilmiştir. Daha sonra Mar'ın ise, geniş çaplı molekül dinamiği simülasyonlarında kullanmak üzere Rapaport'un veri yapılarını sadeleştirme ve performansını artırma çalışmaları yapmıştır.

Mar'ın 'ın Yerel Minimum Algoritması[2]:

Mar'ın olay listesi için yapılan hesaplamaları azaltmayı amaçlar. Bunun için bir yığın mekanizması kullanır. Ancak makalesinde bu yığından tamamlanmış ikili ağaç yapısı olarak basheder.

Gelecek olay listeleri, zaman sırasına konulduğunda, yerel minimum algoritması (LMA), her molekülle ilgili minimum olay zamanını tamamlanmış ikili ağaç (CBT) yapısına koyar.

- Her aşamada bu ağacın kök düğümündeki olayı işler.
- Gelecek olayları tekrar hesaplayıp yeni düğümleri oluşturur.
- Oluşturulan yeni düğümlerle CBT güncellenir.
- Aynı işlemler tekrar edilir.

Burada Mar'ın'ın kullandığı CBT yapısı aslında, en alt seviyesi "id"lerine göre sıralı olarak tutulan moleküllerden oluşan bir yığın yapısıdır. Bu yığının üst kısımları ise daha küçük olay zamanına sahip olan molekülün "id"si daha yukarıda olacak şekilde molekül idlerinden oluşmaktadır. Dolayısıyla bu yığının eleman sayısı sülasyon başından sonuna kadar aynıdır. Bu sayede yığına eleman ekleme ve çıkarma işlemi yapılmaz. Sadece yeni hesaplanan olay zamanlarına göre en alt seviyedeki moleülleri tutan düğümler hariç üst seviye düğümler güncellenir. Böyle bir yapı sayesinde en küçük olay zamanına sahip olan moleküle $O(1)$ zamanda erişilir. Yeni bir olay hesabı sonrası yığını güncellemek ise $O(\log N)$ maliyetindedir.

1995 yılında Mar'ın ve Cordero olay temelli moleküler dinamik simülasyonlarında kullanılan farklı arama algoritmalarını sistematik olarak karşılaştırdılar[3]. Neticede CBT'nin verimi en yüksek bulunur. Eğer molekül sayısı arttırılırsa, CBT'nin performansı diğer arama algoritmalarına göre oldukça daha hızlıdır.

Sonuç olarak CBT ve BST karşılaştırıldığında;

- Karmaşıklıkları aynı yani $O(\log N)$ 'dir.
- Fakat CBT algoritması BST ye göre çok daha basittir. Ve CBT daha az hafızaya ihtiyaç duyar ve gerçek performansı daha iyidir.

4 ÇOK HÜCRELİ YAKLAŞIMA GETİRİLEN YENİLİKLER

Daha önce ifade edildiği gibi, olay temelli molekül dinamiği simülasyonlarında çok hücreli bir yaklaşım kullanılarak büyük performans kazanımları sağlanabilir. Bu yaklaşımın sağladığı başlıca kazanım, moleküller arası çarpışma yani, INTERMOLECULAR_COLLISION olaylarının hesaplamalarına getirdiği yenilikten kaynaklanmaktadır. Bu yenilik bir molekül için çarpışma ihtimali olan moleküllerin arandığı araştırma alanını küçültülmesiyle sağlanmıştır. Her molekül, sadece, bulunduğu hücrenin içindeki moleküllerle ve bulunduğu hücrenin komşu hücrelerindeki moleküllerle çarpışıp çarpışmayacağını kontrol eder. Genel kabuller ışığında ortamın dikdörtgenler prizması şeklinde olduğu ve yüzeylerine paralel olarak daha küçük prizmalara ayrıldığı durumda her hücrenin 26 tane komşu hücresi olacaktır. Bu de demek oluyor ki, bir molekül çarpışma kontrolü yaparken kendi içinde bulunduğu hücre ile birlikte toplam 27 hücre içindeki molekülleri kontrol etmelidir. Bu tez çalışması ile getirdiğimiz yenilik ise tam olarak bu noktadadır. Eğer her hücre için, hücrelerin içindeki ortalama molekül sayısı sabit tutularak, bir hücreye komşu hücre sayısı azaltılabilirse, genel çok hücreli yaklaşımdan daha fazla performans sağlanabilir. Bu iyileştirmeyi elde etmek için ise farklı hücre geometrileri ve farklı hücre dizilimlerinin uygulanması gerekmektedir.

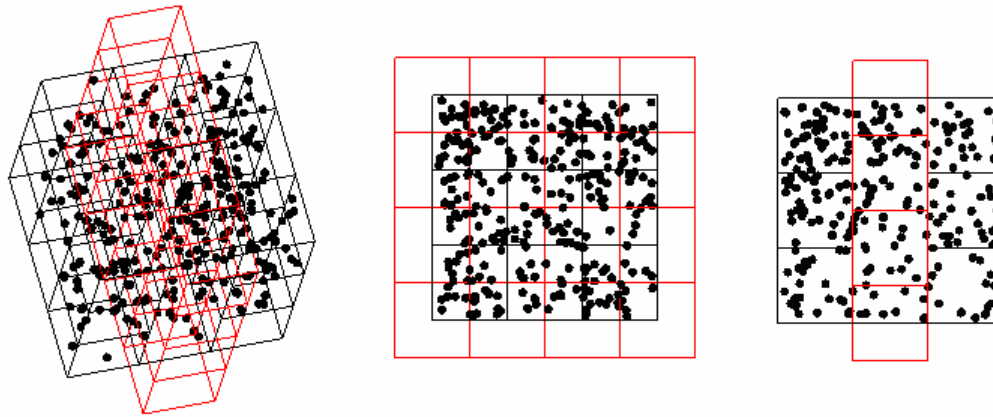
4.1 Uygulanan Hücre Tipleri ve Etkileri

Bu tez kapsamında çok hücreli, olay temelli molekül dinamiği simülasyonlarını hızlandırmak amacıyla, ortamı bölümleyen hücre tipi değiştirilerek, olay hesaplama algoritmasının(Şekil5.4) 1. ve 4. adımlarında hesaplama zamanından kazanç sağlamak hedeflendi. Bu nedenle her hücre için komşu sayısını azaltıldı ve olay hesaplamalarındaki maliyet azaltılmasına rağmen, simülasyonun hiç bir olay kaçırılmadan devam etmesi sağlandı.

Öncelikle, ortam klasik yaklaşımda olduğu gibi alt dikdörtgenler prizmaları şeklinde hücrelere ayrıldı. Bu oluşumda kenarlardaki hücrelerin sınırlarının, ortamın duvarlarına bitişik olması sağlandı. Bu sayede klasik yaklaşımın performansı ile yapılan

uygulamaların performansı karşılaştırılacak ve iyileştirmelerin getirisi görebilecekti. Klasik yaklaşımdaki hücre dağılımı Şekil 3.1’de görülebilir.

Klasik yaklaşımın ardından uygulanan hücelere ayırma şeklinde ise yine dikdörtgenler prizmaları kullanıldı ancak bu defa hücrelerin dizilimi değiştirildi. Z ekseni boyunca aynı şekilde alanlara ayrıldı fakat x ve y eksenleri boyunca klasik ayırma göre hücre büyüklüğünün yarısı kadar kaydırıldı. Bu sayede komşu hücre sayısı 26 dan 16 ya kadar düşürülmüş oldu ki bu büyük bir düşüştür. Bu şekildeki hücre yerleşiminin görünümü Şekil 4.1’de görülebilir.



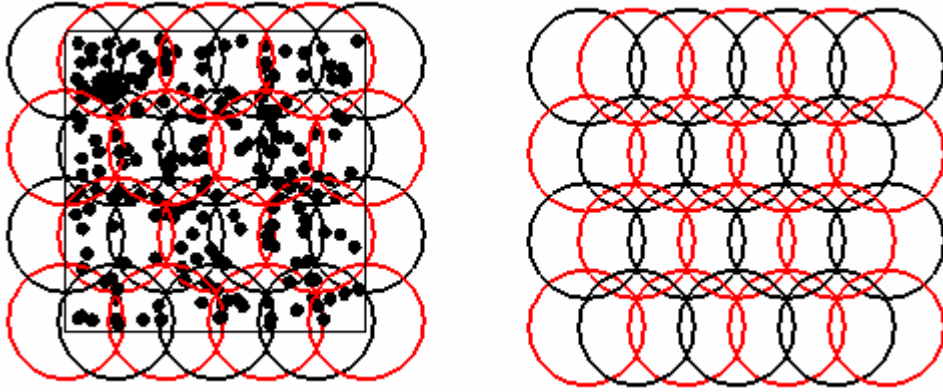
Şekil 4.1 Uygulanan alternatif hücre yerleşimi - HücreTipi2

Bu tip hücre diziliminin performansı ile klasik yaklaşımınki karşılaştırıldığında oldukça yaklaşık %35 oranında performans artışı görülmektedir. Deneysel sonuçlar aşağıdaki bölümde görülebilir.

Daha az komşuluk sağlamak üzere bir başka hücelere ayırma formatı olarak ise küre şekli düşünüldü. Genel olarak hesaplamaları basite indirmek için daha basit hücre yapıları kullanmaya çalışıldı. Küre ise bu bakımdan oldukça uygundu. Hücre geçişi hesaplamaları küre yapısında kolayca hesaplanabilir. Sadece molekülün merkezinin hücrenin merkezine olan uzaklığının hücrenin yarıçapından küçük olup olmadığının kontrolü yeterlidir. Ancak tüm ortamı hiç boşluk kalmayacak şekilde kürelere ayırmak için kürelerin bir miktar kesişmesi gereklidir. Fakat, küreleri yerleştirdiğimizde görüldü ki, ya komşuluk sayısı azalmıyor ya da hücreler arasında boşluk kalıyor.

Son olarak ise silindir biçimli hücre tipi denendi. Silindir şeklinden dolayı kürede olduğu gibi, ortamın tüm hacmini kaplaması için silindirlerin bir miktar kesişmesi gerekir. Yapılan çalışmalar sonunda hücreler, Şekil 4.2’de gösterildiği şekilde dizildiğinde komşuluk sayısı 16 dan 14 e indirilebilir ve hücrenin şeklinden dolayı hücre geçişleri yine kolayca kontrol edilebilir. Bu durumda teorik olarak beklenen hızlanma %44 civarındadır. Ancak pratikte bu artış %40 civarında olmuştur.

Silindir tipli hücrelerin kesişmesinden dolayı, kesişen bölgelerdeki moleküllerin hangi hücreye ait olacağı problemi doğar. Bu sorun ise, bir molekül bir hücreden çıkana kadar diğer hücreye geçmiş sayılmaz kabulüyle ele alınabilir. Bunun olay işleme açısından herhangi bir yan etkisi yoktur. Çünkü molekül herhangi bir çarpışma olayı icra etmek için, uygun pozisyona gelene kadar hücre geçişi yapacaktır.



Şekil 4.2 Uygulanan silindir biçimli hücre şekli - HücreTipi3

Uygulanan bu hücre tiplerinin performansa olan etkileri [DENEYLER](#) bölümünde ayrıntılı olarak anlatılacaktır.

5 KODLAMA DETAYLARI

Öncelikle, çalışmalara temel oluşturması açısından klasik, olay temelli, çok hücreli bir molekül dinamiği simülasyonu geliştirildi. Bu simülasyonda moleküllerin içinde bulunduğu ortamın kapalı ve dışarıdan herhangi bir kuvvet etkisinde olmadığı kabul edildi. Moleküllerin tek atomdan oluştuğu ve moleküller arası çarpışmaların anlık ve esnek olduğu, çarpışan moleküllerin kimyasal reaksiyona girmediği varsayıldı. Böylelikle ortamın toplam enerjisi ve ortamdaki molekül sayısının sabit olması sağlandı.

Simülasyon başlamadan önce moleküllerin ve ortamın nitelikleri belirlendi. Bu koşullar altında, oluşacak her hücre ortalama sekiz moleküle sahip olacak şekilde, ortam hücrelere bölündü. (Sekiz değeri daha önceki çalışmalarda çok hücreli bir molekül dinamiği simülasyonunda olması gereken optimum molekül sayısı olarak hesaplanmıştır[1]). Ortamın konumu ve şekli ile, moleküllerin simülasyonun başında sahip olacakları vektörel ilk hızlar ve ortamın konumuna göre belirlenmiş ilk konumlar verildi. Yaklaşık her sekiz molekül bir hücreye atanmak koşuluyla moleküller ile hücreler ilişkilendirildi. İlkendirme işlemi ileriki bölümlerde detaylı ele alınacaktır.

Bu ilkendirmelerden sonra, olay işleme döngüsü başlatıldı. Her molekülün tek tek yapması muhtemel ilk olayları belirlendi. Her molekül için muhtemel hücre geçiş zamanları, moleküller arası çarpışma zamanları, ve molekül bir sınır hücresi içindeyse duvara çarpma zamanları hesaplanıp, bunlar arasında en yakın zamanlı olan olay moleküle molekülün ilk olayı olarak atandı. Tüm moleküllerin ilk olayları hesaplandıktan sonra, her hücrenin içindeki moleküllerin ilk olayları arasından en küçüğü seçilerek o hücrenin ilk olayı olarak atandı. Bu seçim işlemlerini hızlı bir şekilde yapmak amacıyla kullanılan veri yapılarından ileriki bölümlerde bahsedilecektir.

Her hücrenin ilk olayı bu olayı gerçekleştirecek molekül belli olduktan sonra simülasyon başlatılır. Simülasyon gerçekleşecek ilk olayı işler ve bu olaydan etkilenen hücre ve molekülleri güncelledikten sonra bir sonraki olaya karar verir ve bu olayı işleme geçer. Simülasyon zamanı dolana kadar ya da ortamda işlenecek hiç olay kalmayana (ki

bu çok küçük bir ihtimaldir ve moleküllerin hepsinin hızlarının sıfırlanması anlamına gelmektedir) kadar bu olay işleme döngüsü devam eder.

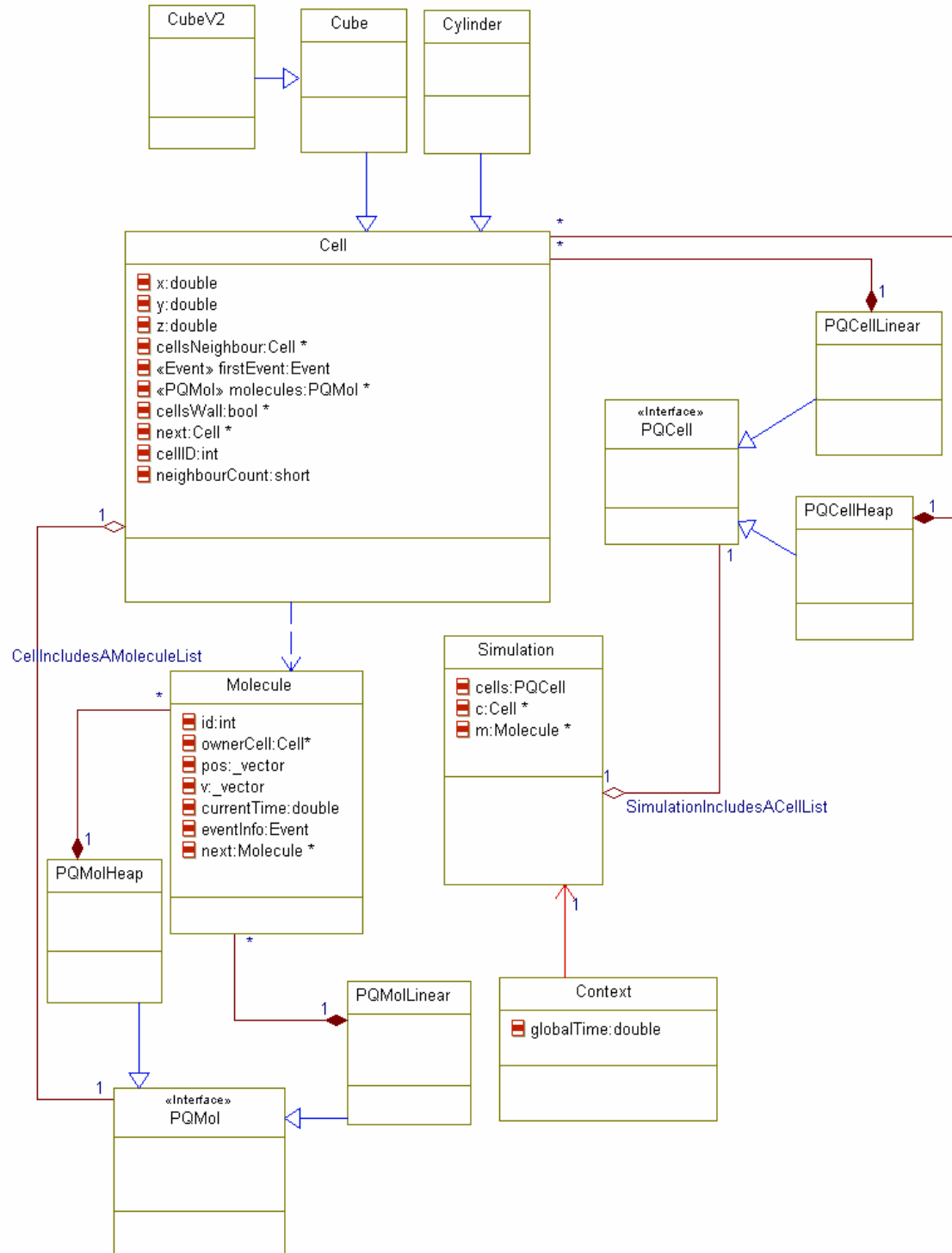
Tüm bu olay işleme döngüsünü etkili bir biçimde takip edebilmek amacıyla, simülasyon boyunca gerçekleşen tüm olaylar, ortam moleküllerin durumları görselleştirilmiştir. Görselleştirmenin detayları takip eden bölümlerde anlatılacaktır.

Getirilen asıl yenilik hücre şekil ve yerleşimlerine olduğundan ilklendirme, olay işleme döngüsü, görselleştirme kısımları tüm farklı hücre bölünümleri için genel manada ortak özelliktedir.

Kodlama detayları; sınıf şemasının verilmesiyle başlayıp, ilklendirme, kullanılan veri yapıları, olay hesaplama işlemi, olay işleme döngüsü ve görselleştirme bölümleriyle devam etmektedir.

5.1 Sınıf Şeması

Geliştirilen uygulamaya ait sınıf şeması aşağıdaki grafikte görülebilir. Bu uygulamada, hücre tipleri “Cell” adı verilen ana bir sınıftan türerler. Küp hücre tipi “Cube” isimli sınıftan oluşur. Bu sınıfın türemiş bir versiyonu ise “CubeV2”dir. Farklı dizilimli küplerden oluşan hücre yapısında hücreler, “CubeV2” sınıfından oluşur. Silindirik hücreleri ise “Cylinder” adlı sınıftan oluşmaktadır. Simülasyondaki moleküller “Molecule” adlı sınıfın objeleridir. “Simulation” isimli sınıf, simülasyonun kontrolünün yapıldığı sınıftır. Şekildeki diğer sınıflar ise simülasyondaki sıralı liste ya da heap yapılarını oluşturmak amaçlı geliştirilmiş sınıf veya arayüzlerdir.



Şekil 5.1 Sınıf Şeması

5.2 İklendirme

İklendirme işlemi simülasyon elementlerinin genel özelliklerinin iklendirmesidir. Bu da ortamın genel simülasyonun iklendirmesi, hücrelerin iklendirmesi ve moleküllerin iklendirmesi olarak sıralanır. Simülasyon başlamadan önce moleküllerin içinde bulundukları kapalı kutu hücrelere ayrılır. Burada dikkat edilmesi gereken konu hücrenin şekli ve yerleşimidir. Hücre geometrisini belirlemede önemli olan faktör hücrelerin birbirlerine olan yakınlığıdır. Hücrelerin dış sınırları birbirine değmelidir ve şekil olarak da küreye yakın olmalıdırlar. Dış sınırları birbirine değen hücreler muhtemel çarpışma eşleri olarak adlandırılır.

Simülasyonun İklendirmesi

Simülasyonun iklendirmesi simülasyondaki hücre sayısının belirlenmesi, hücre tipinin belirlenmesi ve her hücrenin sahip olacağı molekül sayısının belirlenmesinden oluşur. Bunlara ilaveten, ortamın büyüklüğü, ortamın pozisyonu ve hücrelerin büyüklüğünün belirlenmesini kapsar.

Hücrelerin İklendirilmesi

Hücrelerin iklendirilmesi hücre pozisyonlarının belirlenmesi, her hücrenin komşuluklarının belirlenmesi ve hücrenin simülasyon ortamına bitişik olan ya da başka bir deyişle simülasyon ortamının duvarlarından birini içeren yüzeylerinin belirlenmesinden oluşur. Ayrıca tüm hücrelerin ilk olayları boş olarak atanır.

Moleküllerin İklendirilmesi

Moleküllerin iklendirilmesi molekül yarıçaplarının belirlenmesi ve moleküllerin pozisyonlarına göre ait oldukları hücrelere atanmasından oluşur. Her molekül sadece ve sadece bir hücreye atanır. Bir molekülün hangi hücreye ait olacağı o molekülün merkezinin nerede olduğuna göre belirlenir. Bunlara ilaveten iklendirme işlemi, moleküllere ilk doğrusal hızların verilmesini de içerir. Bizim uygulamamızda hız ve pozisyon gibi iklendirmeler rastgele(random) olarak yapılmıştır. Ayrıca tüm moleküllerin ilk olayları başlangıçta boştur.

5.3 Kullanılan Veri Yapıları

Simülasyon boyunca, her aşamada, en yakın zamanlı olaya sahip olan molekülü içinde barındıran hücrenin belirlenmesi gerekmektedir. Bu sebeple her aşamada bir minimum seçimi gereklidir. Bu sebeple simülasyondaki hücrelerin sıralı tutulmasına ilaveten her hücre içindeki moleküller de muhtemel ilk olaylarının zamanlarına göre sıralı tutulmuştur. Her hücrenin ve simülasyonun kendilerine ait sıralı listeleri vardır. Hücreler için bu listeler hücrenin sahip olduğu molekülleri, bu moleküllerin gelecek muhtemel olaylarının oluş sırasına göre sıralı olarak tutar. Simülasyon için de benzer şekilde, ortamdaki hücreleri, bu hücrelerin muhtemel ilk olaylarının oluş sırasına göre sıralı olarak tutar. Sıralı listelere elemanlar minimum olay zamanlarına göre sıralı olarak yerleştirilmiştir. Yapılan implementasyonda sıralı listeler en basit uyarlanabileceği veri yapıları olan “bağlı listeler” olarak implement edilmiştir. Ancak yığın veri yapısının sistemin performansı açısından daha iyi olabileceği düşünülmüş ve ileriki çalışmalarda bu yapının kullanılması öngörülmüştür.

Olay Veri Yapısı

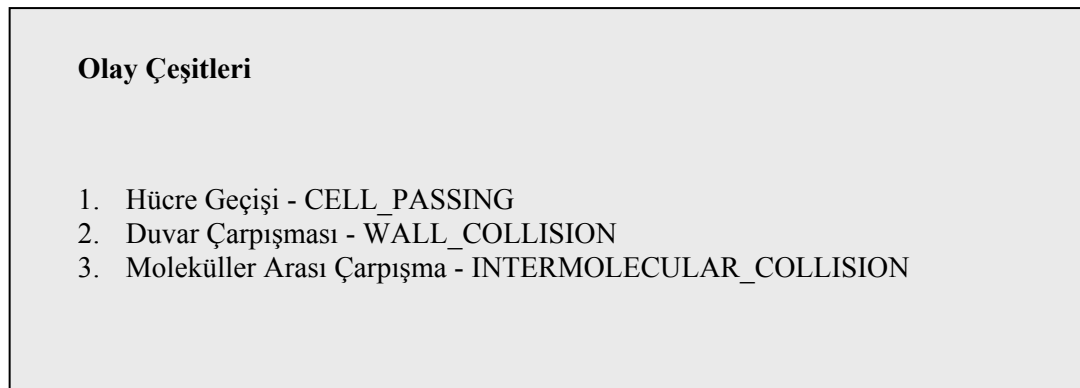
```
struct Event
{
    double eventTime; /* Olay zamanını tutuyor.*/
    short eventType; /* Olay tipini tutuyor.*/
    Molecule * firstEventIsRelatedWithThisMolecule;
    /* Çarpışılacak molekülü tutuyor.*/
    Cell * whichCell; /*Gececeği hücreyi */
    short whichWall; /* Hangi duvara carpacagini tutuyor.*/
};
```

Şekil 5.2 Olay veri yapısı

Olaylar bir olay veri yapısı içinde saklanır. Bu veri yapısının elemanları ise olayın zamanı, olayın tipi, olayla ilişkili hücre, olayla ilişkili duvar ve olayla ilişkili moleküldür. Olayla ilişkili hücre bilgisi, hücre geçişi olaylarında geçilecek hücreyi tutmak için kullanılır, eğer olay hücre geçişi değilse bu bilginin değeri boştur. Eğer olay tipi moleküller arası çarpışma ise olayla ilişkili molekül bilgisi tutulur aksi takdirde bu bilginin değeri boştur. Yine benzer şekilde olay tipi duvara çarpma ise olayla ilişkili duvar bilgisi tutulur aksi halde bu bilgi de boştur. Olay veri yapısı Şekil 5.2 da görülebilir.

5.4 Olay Hesaplama İşlemi

Simülasyonun en başında, gerçekleştirilecek olaylar şu sırayla hesaplanır. Önce pozisyonlarına göre bir dizinde tutulan her bir hücre tek tek ele alınır. Her hücrenin içindeki oluşum sıralarına göre belli “id”ler atanmış olan moleküller, bu Idlerin sırasına göre, tek tek olay hesaplama işlemine girer. Olay işleme döngüsü ise 3 adımdan oluşur. Bunlar sırasıyla Şekil 5.3’de belirtilen olay çeşitlerinin kontrolüdür.



Şekil 5.3 Olay çeşitleri

Hücre geçişi olayı, çok hücreli simülasyondan kaynaklanan bir olaydır. Çok hücreli yaklaşımda her molekül belirli bir hücreyle ilişkilendirmiştir, yani o hücre içinde yer alır. Bu hücrenin komşusu olan başka bir hücreye geçebilmesi demek, merkezinin bulunduğu hücrenin sınırına ulaşmış olması demektir. Kısaca, molekülün merkezi ile hücrenin sınırları arasındaki mesafenin, molekülün o yöndeki hız bileşenine bölünmesiyle, molekülün ait olduğu hücreden çıkıp komşu bir hücreye geçme zamanı bulunur. Geçilecek komşu hücre ise bulunan değerlerin en küçüğünün olduğu yöndeki hücredir.

Şayet, ortam x , y ve z koordinat eksenlerine paralel bir şekilde yerleştirilir ve yine bu eksenlere paralel olarak hücrelere ayrılırsa, sadece molekülün x , y ve z yönündeki hız bileşenlerinin doğrultusunda yer alan hücre sınırlarının kontrolü yeterlidir. Hız bileşenleri yönündeki muhtemel hücre geçiş zamanları hesaplandıktan sonra, bunlar arasından en küçüğü seçilir ve o molekülün hücre geçiş zamanı olarak saklanır. Bunun yanında geçilecek hücre bilgisi de saklanmalıdır.

Bir molekül için hücre geçiş zamanı belirlendikten sonra, eğer molekülün içinde bulunduğu hücre ortam duvarlarından biriyle bitişik ise, benzer şekilde duvara çarpma zamanı bulunur. Ancak bu kez hesaplanacak mesafe hücrenin merkezi ile ortam duvarı arasındaki mesafe değildir. Bu kez molekülün yarıçapı işin içine girer ve molekülün dış yüzeyinin ortam duvarına yani duvara bitişik olan hücre sınırına ne zaman ulaşacağı bulunur. Bu ise, molekülün merkezi ile yarıçap büyüklüğünün toplamıyla elde edilen değer ile hücrenin kontrol edilecek duvara bitişik olan sınırı arasındaki mesafenin, molekülün o yöndeki hız bileşenine bölünmesiyle elde edilir. Muhtemel çarpışmanın yapılacağı ortam duvarı bulunan değerlerin en küçüğünün olduğu yöndeki duvardır. Şayet, ortam x , y ve z koordinat eksenlerine paralel bir şekilde yerleştirilir ve yine bu eksenlere paralel olarak hücrelere ayrılırsa, sadece molekülün x , y ve z yönündeki hız bileşenlerinin doğrultusunda yer alan ve ortam duvarlarından biriyle bitişik olan hücre sınırlarının kontrolü yeterlidir. Hız bileşenleri yönündeki muhtemel duvara çarpma zamanları hesaplandıktan sonra, bunlar arasından en küçüğü seçilir ve o molekülün duvara çarpma zamanı olarak saklanır. Buna ilaveten çarpılacak duvarın hangisi olduğu bilgisi de saklanmalıdır.

Son olarak, bir molekülün moleküller arası çarpışma yapma ihtimali olup olmadığının ve muhtemel çarpışma eşinin belirlenmesi için, başka moleküllerle yapması muhtemel çarpışmaların kontrolü gereklidir. Bunun için öncelikle molekülün içinde bulunduğu hücrenin içinde yer alan diğer moleküllerle çarpışıp çarpışmadığı kontrol edilir. Şayet muhtemel bir çarpışma yakalanırsa, olayın zamanı ve ilgili molekül kaydedilir. Daha sonra molekülün içinde bulunduğu hücrenin komşu hücrelerinde yer alan moleküllerle çarpışıp çarpışmayacağı kontrol edilir. Muhtemel böyle bir çarpışmaya rastlanırsa bu çarpışmaya ait ilgili molekül, ilgili hücre ve olay zamanı bilgileri

kaydedilir. Daha sonra belirlenen bu iki olay bilgilerinden küçüğü, yada sadece tek bir tanesi belirlenmişse o belirlenen olay, o molekülün muhtemel moleküller arası çarpışmasını belirten olay olarak kaydedilir. İki molekül arasındaki çarpışma ise moleküllerin merkezleri arasındaki mesafe molekül çapı büyüklüğüne eşit olabiliyor mu, eşit olabiliyorsa ne zaman sorularının çözümüyle elde edilir. İki molekülün çarpışma hesabı ise şöyle yapılır:

İlk molekülün konum vektörü k_1 , ikinci molekülün k_2 olsun. İlk molekülün hız vektörü v_1 , ikincinin v_2 olsun. Çarpışma zamanları da t olsun. Bu durumda $k_2 + v_2 * t$, $k_1 + v_1 * t$ elimizde var. Bunların içinden t yi çekeceğiz.

$$A = k_1 + v_1 * t;$$

$$B = k_2 + v_2 * t; \quad \text{dersek:}$$

$$t = (A - k_1) / v_1;$$

$$t = (B - k_2) / v_2; \quad \text{olur.}$$

A-B vektörünün uzunluğu da $R_1 + R_2$ (yarıçap)dır.

$$A = (m_1x + v_1x * t)x + (m_1y + v_1y * t)y + (m_1z + v_1z * t)z;$$

$$B = (m_2x + v_2x * t)x + (m_2y + v_2y * t)y + (m_2z + v_2z * t)z;$$

$$||A - B|| = R_1 + R_2 =$$

$$= \sqrt{(m_1x + v_1x * t - m_2x - v_2x * t)(m_1x + v_1x * t - m_2x - v_2x * t) +$$

$$(m_1y + v_1y * t - m_2y - v_2y * t)(m_1y + v_1y * t - m_2y - v_2y * t) +$$

$$(m_1z + v_1z * t - m_2z - v_2z * t)(m_1z + v_1z * t - m_2z - v_2z * t));$$

$$= \sqrt{(m_1x - m_2x + t(v_1x - v_2x)) * (m_1x - m_2x + t(v_1x - v_2x)) +$$

$$((m_1y - m_2y + t(v_1y - v_2y)) * (m_1y - m_2y + t(v_1y - v_2y))) +$$

$$((m_1 z - m_2 z + t^*(v_1 z - v_2 z)) * (m_1 z - m_2 z + t^*(v_1 z - v_2 z)));$$

$$= ||(k_1 - k_2) + t^*(v_1 - v_2)||$$

P vektörü iki konum vektörünün farkı olsun.

$$P = k_1 - k_2;$$

$$P_x = m_1 x - m_2 x;$$

$$P_y = m_1 y - m_2 y;$$

$$P_z = m_1 z - m_2 z;$$

V vektörü iki hız vektörünün farkı olsun.

$$V = v_1 - v_2;$$

$$V_x = v_1 x - v_2 x;$$

$$V_y = v_1 y - v_2 y;$$

$$V_z = v_1 z - v_2 z;$$

Bu durumda formül şuna döner:

$$R_1 + R_2 = ||(P + t * V)||;$$

$$R_1 + R_2 = \text{sqrt}((P_x + t * V_x) * (P_x + t * V_x) + (P_y + t * V_y) * (P_y + t * V_y) +$$

$$(P_z + t * V_z) * (P_z + t * V_z));$$

$$\Rightarrow (R_1 + R_2) * (R_1 + R_2) = ((P_x + t * V_x) * (P_x + t * V_x)) + ((P_y + t * V_y) * (P_y + t * V_y))$$

$$+ ((P_z + t * V_z) * (P_z + t * V_z));$$

$$\Rightarrow (R_1 + R_2) * (R_1 + R_2) = (P_x * P_x + 2 * P_x * t * V_x + t * V_x * t * V_x)$$

$$+ (P_y * P_y + 2 * P_y * t * V_y + t * V_y * t * V_y) + (P_z * P_z + 2 * P_z * t * V_z + t * V_z * t * V_z)$$

$K = (R1+R2)*(R1+R2)$ olsun,

$L = ||P|| * ||P||$ olsun,

$M = ||V|| * ||V||$ olsun.

Bu durumda formül şuna döner:

$K = L + t*t * M + 2*t(Px*Vx + Py*Vy + Pz*Vz);$

P vektörü ile V vektörünün “dot product”ının iki katı N olsun.

Bu durumda formül şöyle olur:

$M * t^2 + N * t + L = K;$

$M * t^2 + N * t + L - K = 0;$

discriminant = $N^2 - 4*M*(L-K)$ olur, kökler ise : $(- N \pm \sqrt{\text{discriminant}}) / 2*M$ olur.

Bu değer de çarpışma zamanına eşittir.

Simülasyonda her molekül için hücre geçişi, molekül çarpışması ve duvar çarpışması değeri ayrı ayrı hesaplanır ve yine her molekül için bu üç olay değerinin en küçük olay zamanına sahip olanı o molekülün ilk olayı olarak atanır. İlk olayı belirlenen her molekül, ait olduğu hücrenin öncelik kuyruğuna, olay zamanlarına göre küçükten büyüğe sıralı olacak şekilde eklenir. Bundan sonraki yapılacak ilk iş hücrenin ilk olay zamanının hesaplanmasıdır ki bu da öncelik kuyruğu veri yapısının kullanılmasından dolayı $O(1)$ lik bir işlemdir. Her hücrenin olay önceliklerine göre dizilmiş molekül listesindeki ilk molekülün olayı hücrenin de ilk olayıdır ve o şekilde atanır.

Molekülün İlk Olayını Hesaplama Döngüsünün kodumsu hali Şekil 5.4 de görülebilir.

1. Molekölün bulunduğu hücrenin komşu hücrelerinden hangisine ve ne zaman geçebileceğini hesapla. Olay zamanını olay tipini kaydet.
2. Molekül sınır hücrelerinden birinde yer alıyorsa muhtemel duvar çarpışmalarını hesapla. İlgili duvarı, çarpışma zamanını ve olay tipini kaydet.
3. İlgili molekülle aynı hücrede bulunan moleküller arasındaki muhtemel çarpışmaları hesapla. Çarpışılacak molekülü, çarpışma zamanını, ve olay tipini kaydet.
4. İlgili molekülün bulunduğu hücrenin komşu hücrelerinde bulunan moleküller arasındaki muhtemel çarpışmaları hesapla. Çarpışılacak molekülü, çarpışma zamanını, ve olay tipini kaydet.
5. Minimum pozitif olay zamanına sahip olan olayı seç. Eğer böyle bir olay varsa bu olayı molekülün ilk olayı olarak ata.

Şekil 5.4 Bir olay hesaplama detayları

Her hücrenin ilk olayı hesaplanırken bir yandan da simülasyonun öncelik kuyruğuna hücreler ilk olaylarına göre eklenir. Bu sayede simülasyon başlamadan önce tüm hücreler ilk olaylarına göre sıralı bir biçimde elde edilmiş olur.

Simülasyon başlamadan önceki olay hesaplama işlemi bittikten sonra, simülasyon, olaylarına göre sıralı olarak tuttuğu hücrelerin ilkinin ilk olayını işler. Daha sonraki olay hesaplama işlemleri işleme tutulacak her molekül ve her hücre için yukarıdaki adımlardan oluşur. Olay işleme döngüsü ise aşağıdaki bölümde açıklanacaktır.

5.5 Olay İşleme Döngüsü

Yukarıda da bahsedildiği gibi üç tip olay tanımlıdır. Bunlardan ilki ve en basiti bir molekülün bir hücreden diğerine geçmesidir. Bu olayın gerçekleşmesi molekülün yeni ait olacağı hücreye atanması, yeni yerinin belirlenmesi ve molekül içi zamanının olay zamanına getirilmesi şeklinde olur. İkinci tip olay ise iki molekülün birbirleriyle çarpışmasıdır. Bu olayın gerçekleşmesi için bu ilgili moleküllerin durumlarının güncellenmesinin yanında bu olayın sonucundan etkilenmesi muhtemel moleküllerin belirlenmesi ve onların durumlarının da güncellenmesi gerekmektedir. Moleküllerin durumlarının güncellenmesi ise moleküllerin yeni hızlarının, yeni yerlerinin ve molekül içi zamanlarının güncellenmesi şeklinde olur. Son tip olay ise bir molekülün kutunun duvarlarından birine çarpmasıdır ki bunun sonucunda da molekülün yeri, hızı ve molekül içi zamanı değişir.

Tüm bu olaylar gerçekleşdikten sonra, bu olaylarda adı geçen moleküllerin gelecek olayları hesaplanır ve öncelik listesine konulur. Daha sonra aynı işleme simülasyon zamanı boyunca devam edilir.

Olay işleme döngüsünün olay hesaplama işleminin çıktılarını da içeren kodumsu hali Şekil 5.5 da görülebilir.

1. Her molekülün ilk olayını hesapla.
 - a. Olayın tipini hesapla.
 - b. Olayın zamanını hesapla.
 - c. Olay moleküller arası çarpışma ise, bu çarpışmaya katılacak diğer molekülü bul.
 - d. Hücrenin öncelik sırasına göre düzenlenmiş olan molekül listesinin uygun yerine molekülü olay zamanını baz alarak yerleştir.
2. Her hücrenin ilk olayını, sahip olduğu molekül listesinin ilk molekülünün, ilk olayı olarak ata.
3. Simülasyonun sahip olduğu hücrelerin oluşturduğu öncelik kuyruğunu hücrelerin ilk olaylarına göre güncelle.
4. Simülasyonun öncelik kuyruğunun en başındaki hücrenin olayını işle.
5. Olaydan etkilenen moleküllerin gelecek olaylarını hesapla.
6. Olaydan etkilenen hücrelerin moleküllerini tutan öncelik kuyruklarını güncelle.
7. Simülasyonun hücreleri tutan öncelik kuyruğunu güncelle.
8. 4. adıma geri dön.

Şekil 5.5 Olay işleme döngüsü

5.6 Görselleştirme

Simülasyonda görselleştirme aracı olarak OpenGL kütüphanesi ve OpenGL üzerinde GLUT kütüphanesi kullanıldı. Simülasyon üç boyutlu olarak görüntülendi ve kullanıcıya simülasyonun herhangi bir anında ortamı herhangi bir açıdan izleme fırsatı sunuldu. Moleküller küre olarak ortam da dikdörtgenler prizması olarak modellendi. Hücrelere ayırıştırma şekilleri de ortam üzerine giydirildi. Bu sayede kullanıcının hangi hücrede kaç molekül var hangi molekül nerde duruyor ve hangi olayı gerçekleştiriyor bilgisini ekrandan görebilmesi sağlandı. Simülasyonun herhangi bir aşamada durdurulup farklı açılardan incelenmesi sağlandı.

6 DENEYLER

Simülasyon sırasıyla aşağıdaki durumlarda çalıştırıldı. Her durumun sonucunda alınan değerler hemen takibinde belirtilmiştir.

Simülasyon Akış Değişkeni: (Global_Time) Simülasyonun içindeki akış bu değişken yolu ile kontrol edilmiştir. Aynı molekül sayısı için, eşit sayıda çarpışma yapacak şekilde, farklı hücre tiplerinde simülasyon çalıştırıldığında, tüm hücre tipleri için simülasyon sonunda simülasyon akış değişkeni aynı değere ulaşmalı ve moleküllerin son konumları tutarlı olmalıdır. Hedeflenen çarpışma sayısına ne kadar sürede erişildiği ise simülasyonun performansını gösterecektir.

Molekül Sayısı: Simülasyondaki molekül sayısıdır.

Hücre Tipi: Simülasyonun çalıştırıldığı hücre tipidir.

Simülasyon zamanı: Simülasyon çalıştırıldığında hedeflenen çarpışma sayısına kaç saniyede ulaşıldığı bilgisidir.

Simülasyon deneyleri komşuluk sayısı 26 olan klasik çok hücreli yaklaşım, komşuluk sayısı 16 olan bu yaklaşıma getirilen alternatif hücre dizilimi ve silindir şekilli hücre dizilimi için yapılmıştır. Bu hücre tip ve dizilimleri deney anlatımlarımda sırasıyla Küp, KüpV2 ve Silindir olarak kodlanacaktır.

Yapılan deneylerde simülasyon spacing ratio 3.5, 4, 4.5 ve 5 değerlerinde çalıştırılmıştır. Spacing ratio ise moleküller düzenli olarak ortama yerleştirildiğinde, molekül merkezleri arasındaki mesafenin molekül çapına oranı olarak tanımlanmıştır. Simülasyon 1000, 8000, 27000, 64000 molekül için belirtilen spacing ratio değerlerinin tümünde çalıştırılmış ve alınan sonuçlar karşılaştırmalı olarak aşağıda listelenmiştir. Spacing ratio 4 değeri için ise 125000 molekül için deneyler gerçekleştirilmiştir.

Asıl simülasyon değerlerini toplamadan önce, belirtilen değerler ile çalıştırılmak üzere, moleküller düzgün olarak simülasyon ortamına yerleştirilmiştir. Her molekül ortalama 10 çarpışma yapana kadar simülasyon çalıştırılmıştır ve elde edilen son

konumlar kaydedilmiştir. Daha sonraki aşamada moleküller bu kayıtlı konumlar dosyadan okunmak suretiyle ortama yerleştirilmiş ve simülasyon sonuçları bu aşamada toplanmıştır.

6.1 Elde Edilen Veriler

6.1.1 Veri I

Bu veride, simülasyon, tanımlı spacing ratiolarında, belirlenen miktarlarda, farklı hücre tipleri ile çalıştırılmak suretiyle elde edilen simülasyon süreleri listelenmektedir.

Molekül Sayısı: 1000 (Çarpışma Sayısı = Molekül Sayısı*10)	Spacing Ratio			
Hücre Tipi	3.5	4	4.5	5
Küp	47	55	62	73
KüpV2	32	37	42	50
Silindir	26	31	35	42

Tablo 6.1 Molekül Sayısı 1.000 iken gerçekleşen 10.000 çarpışmanın tanımlı spacing ratiolar ile farklı hücre tiplerinde gerçekleşme zamanları

Molekül Sayısı: 8000 (Çarpışma Sayısı = Molekül Sayısı*10)	Spacing Ratio			
Hücre Tipi	3.5	4	4.5	5
Küp	488	563	642	735
KüpV2	322	372	427	489
Silindir	254	291	337	386

Tablo 6.2 Molekül Sayısı 8.000 iken gerçekleşen 80.000 çarpışmanın tanımlı spacing ratiolar ile farklı hücre tiplerinde gerçekleşme zamanları

Molekül Sayısı: 27000 (Çarpışma Sayısı = Molekül Sayısı*10)	Spacing Ratio			
Hücre Tipi	3.5	4	4.5	5
Küp	2012	2355	2735	3153
KüpV2	1383	1641	1907	2225
Silindir	1245	1483	1749	2038

Tablo 6.3 Molekül Sayısı 27.000 iken gerçekleşen 270.000 çarpışmanın tanımlı spacing ratiolar ile farklı hücre tiplerinde gerçekleşme zamanları

Molekül Sayısı: 64000 (Çarpışma Sayısı = Molekül Sayısı*10)	Spacing Ratio			
Hücre Tipi	3.5	4	4.5	5
Küp	8999	11063	13288	16267
KüpV2	7715	9495	11841	14173
Silindir	7510	9290	11359	13679

Tablo 6.4 Molekül Sayısı 64.000 iken gerçekleşen 640.000 çarpışmanın tanımlı spacing ratiolar ile farklı hücre tiplerinde gerçekleşme zamanları

6.1.2 Veri II

Bu veride, simülasyon, tanımlı spacing ratiolarında, farklı hücre tiplerinde belirlenen miktarlarda çarpışma için geçen sürede yapılan hücre geçişi miktarları listelenmektedir.

Molekül Sayısı: 1000 (Çarpışma Sayısı = Molekül Sayısı*10)	Spacing Ratio				
	Hücre Tipi	3.5	4	4.5	5
	Küp	33755	43787	54886	70295
	KüpV2	35884	46878	58439	75042
	Silindir	35084	45716	57651	74203

Tablo 6.5 Molekül Sayısı 1.000 iken gerçekleşen 10.000 çarpışma süresince, farklı hücre tiplerinde yapılan hücre geçişi sayıları

Molekül Sayısı: 8000 (Çarpışma Sayısı = Molekül Sayısı*10)	Spacing Ratio			
Hücre Tipi	3.5	4	4.5	5
Küp	295497	386284	493234	607180
KüpV2	306348	400809	511666	629419
Silindir	292908	383927	493159	609005

Tablo 6.6 Molekül Sayısı 8.000 iken gerçekleşen 80.000 çarpışma süresince, farklı hücre tiplerinde yapılan hücre geçişi sayıları

Molekül Sayısı: 27000 (Çarpışma Sayısı = Molekül Sayısı*10)	Spacing Ratio			
Hücre Tipi	3.5	4	4.5	5
Küp	1052489	1353787	1718240	2119530
KüpV2	1048290	1384327	1756263	2166802
Silindir	1004223	1332130	1698973	2099618

Tablo 6.7 Molekül Sayısı 27.000 iken gerçekleşen 270.000 çarpışma süresince, farklı hücre tiplerinde yapılan hücre geçişi sayıları

Molekül Sayısı: 64000 (Çarpışma Sayısı = Molekül Sayısı*10)	Spacing Ratio			
Hücre Tipi	3.5	4	4.5	5
Küp	2466581	3245008	4127626	5120580
KüpV2	2510854	3301893	4200073	5210539
Silindir	2400722	3175422	4056665	5050205

Tablo 6.8 Molekül Sayısı 64.000 iken gerçekleşen 640.000 çarpışma süresince, farklı hücre tiplerinde yapılan hücre geçişi sayıları

6.1.3 Veri III

Bu veride, simülasyon, tanımlı spacing ratiolarda, belirlenen miktarlarda çarpışma için geçen sürede çalıştırıldığında, yapılan duvar çarpışması miktarları listelenmektedir. Bu miktarlar farklı hücre tipleri için aynıdır.

(Çarpışma Sayısı = Molekül Sayısı*10)	Spacing Ratio			
Molekül Sayısı	3.5	4	4.5	5
1000	8823	11231	14052	17892
8000	34486	44382	56380	68510
27000	77053	99841	125644	153537
64000	136648	176430	221871	274597

Tablo 6.9 Farklı spacing ratio'larda ve farklı molekül sayılarında gerçekleştirilen duvar çarpışmaları

6.1.4 Veri IV

Bu veride, simülasyon, tanımlı spacing ratiolarında, farklı hücre tiplerinde, belirlenen miktarlarda çarpışma için çalıştırıldığında geçen süreler, molekül sayısı bazında listelenmektedir. Bu tablolardan molekül sayısı ile simülasyon zamanı arasındaki ilişki görülebilir.

Spacing Ratio: 3.5	Molekül Sayısı (Çarpışma Sayısı = Molekül Sayısı*10)			
Hücre Tipi	1000	8000	27000	64000
Küp	47	488	2012	8999
KüpV2	32	322	1383	7715
Silindir	26	254	1245	7510

Tablo 6.10 Spacing ratio 3.5 değerinde simülasyon zamanları

Spacing Ratio: 4	Molekül Sayısı (Çarpışma Sayısı = Molekül Sayısı*10)			
Hücre Tipi	1000	8000	27000	64000
Küp	55	563	2355	11063
KüpV2	37	372	1641	9495
Silindir	31	291	1483	9290

Tablo 6.11 Spacing ratio 4 değerinde simülasyon zamanları

Spacing Ratio: 4.5	Molekül Sayısı (Çarpışma Sayısı = Molekül Sayısı*10)			
Hücre Tipi	1000	8000	27000	64000
Küp	62	642	2735	13288
KüpV2	42	427	1907	11841
Silindir	35	337	1749	11359

Tablo 6.12 Spacing ratio 4.5 değerinde simülasyon zamanları

Spacing Ratio: 5	Molekül Sayısı (Çarpışma Sayısı = Molekül Sayısı*10)			
Hücre Tipi	1000	8000	27000	64000
Küp	73	735	3153	16267
KüpV2	50	489	2225	14173
Silindir	42	386	2038	13679

Tablo 6.13 Spacing ratio 5 değerinde simülasyon zamanları

6.1.5 Veri V

Bu veri, simülasyon, spacing ratio 3.5 değerinde, silindir hücre tipinde, 8.000 molekül için 80.000 molekül çarpışması süresince, farklı silindir derinliklerinde çalıştırılarak elde edilmiştir. Bu tablolardan silindir hücre derinliği ile simülasyon zamanı arasındaki ilişki görülebilir. x ve y eksenleri boyunca hücre büyüklüğü 7 olarak verilmiştir. Bu durumda hücre yarıcapı, kenar büyüklüğü 7X7 olan bir kareyi kapsayacak minimum şekilde belirlenir. Bu değerler sabit tutulrken silindir derinliği 7, 8, 9, ve 10 değerleri için simülasyon çalıştırılır.

X = 7 Y = 7 Molekül Çapı : 1	Spacing Ratio : 3.5 Molekül Sayısı : 8000 (Çarpışma Sayısı = Molekül Sayısı*10)		
Silindir Derinliği	Hücre Geçiş	Duvar Çarpışması	Simülasyon Zamanı
7	292908	34486	385
8	282533	34486	416
9	272370	34486	449
10	261707	34486	486

Tablo 6.14 Spacing ratio 3.5 değerinde, silindir hücre tipinde, farklı silindir derinliklerinde simülasyon zamanları

Hücre geçişi azaldığı halde simülasyon zamanının arttığı görülmüştür. Bunun sebebi ise hücre sayısının azalması ve dolayısıyla bir hüreye düşen ortalama molekül sayısının artmasıdır.

6.1.6 Veri VI

Aşağıdaki tablolar simülasyon 125.000 molekül için, 1.250.000 molekül çarpışması boyunca çalıştırıldığıda elde edilen değerleri göstermektedir.

	Spacing Ratio: 3.5 Molekül Sayısı :125.000 (Çarpışma Sayısı = Molekül Sayısı*10)		
Hücre Tipi	Hücre Geçişi	Duvar Çarpışması	Simülasyon Zamanı
Küp	6403881	275647	34752
KüpV2	6488802	275647	31856
Silindir	6204242	275647	33887

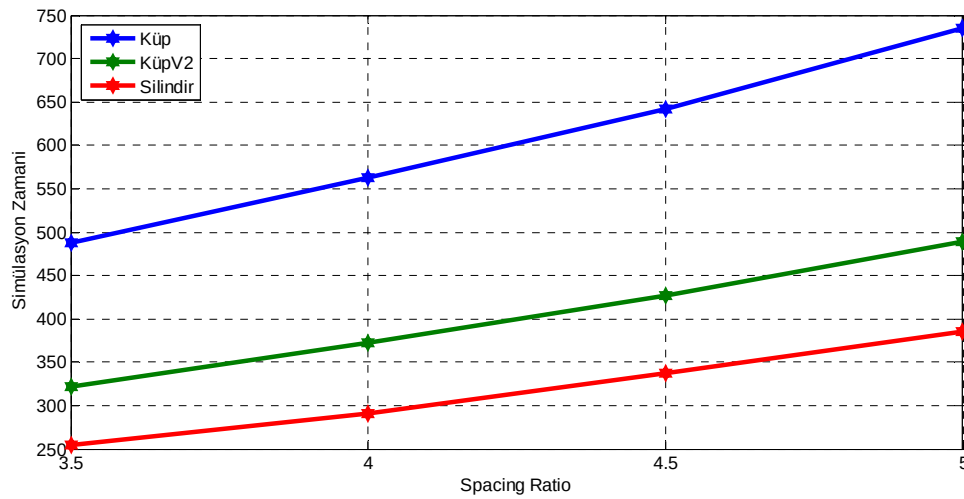
Tablo 6.15 Molekül sayısı 125.000 için simülasyon değerleri

6.2 VerilereDayalı Grafik ve Analizler

Bu bölümde yukarıda verilen verilerin bir kısmı kullanılarak analizler yapılacaktır. Bu analiz sonuçları grafiksel olarak gösterilecektir.

6.2.1 Simülasyon Zamanı

Bu grafikte 8000 molekülün 80.000 molekül çarpışması tamamlayana kadar geçen simülasyon süreleri gösterilmektedir.

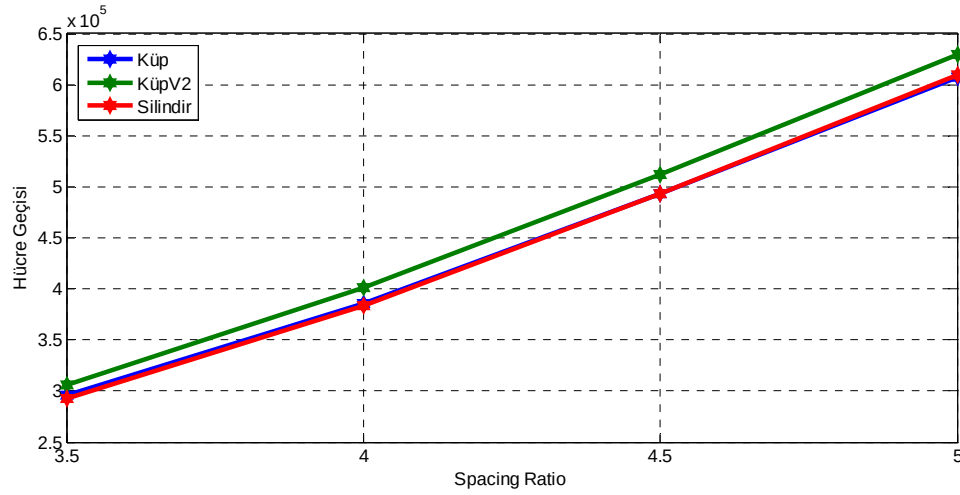


Şekil 6.1 Spacing Ratio X Simülasyon Zamanı

Grafikten görüldüğü üzere spacing ratio arttıkça, yani molekül yoğunluğu azaldıkça, aynı sayıda molekülle aynı sayıda çarpışmaya ulaşana kadar geçen süre tüm hücre tiplerinde artmıştır. Bunun yanında simülasyon zamanları karşılaştırıldığında silindir hücre tipinin en iyi küp hücre tipinin ise en kötü sonucu verdiği görülebilir.

6.2.2 Hücre Geçişi

Aşağıdaki grafikte farklı spacing ratiolarda, 8.000 molekül ile 80.000 molekül çarpışması gerçekleşene kadar simülasyon gerçekleştirilmiş ve bu süre zarfında farklı hücre tiplerinde yapılan hücre geçişi miktarları gösterilmiştir.

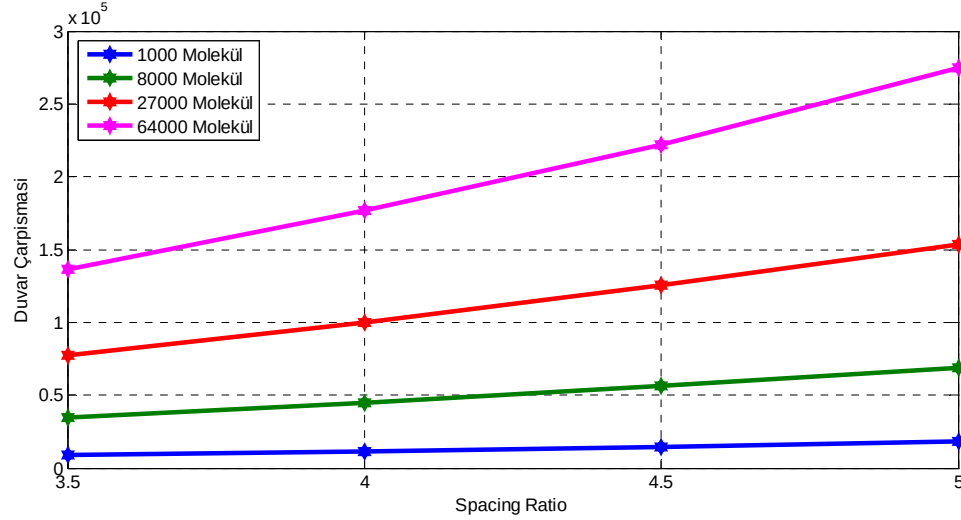


Şekil 6.2 Spacing Ratio X Hücre Geçişi

Grafikte de görüldüğü üzere spacing ratio arttıkça simülasyondaki hücre geçiş olayı miktarı da artmıştır. Farklı hücre tiplerine bakıldığında ise, küpV2 hücre paylaşımının en fazla hücre geçişine neden olduğu küp ve silindir hücre paylaşımının ise yaklaşık aynı sayıda hücre geçişine neden olduğu görülmektedir. KüpV2nin hücre geçiş sayısının küp paylaşımına göre daha fazla olması beklenen bir durumdur. Komşuluk sayısı azaldıkça hücreyi çevlreyen komşu hücrelerin oluşturduğu hacim azalır ve bu nedenle molekül bir çarpışma olayını gerçekleyene kadar yapacağı hücre geçişi miktarı artar. Silindirde durumun farklı olmasının nedeni ise silindir hücrelerinin kesişmesidir. Silindir içindeki bir molekül bir kesişim bölgesine girdiğinde değil ancak ait olduğu silindirin dışına çıktığında hücre geçişi yapmış sayılır. Bu nedenle gerçekleştirilen hücre sayısı en iyi durumdaki ile yaklaşık aynı miktarda azdır.

6.2.3 Duvar Çarpışması

Aşağıdaki grafik simülasyonun, spacing ratio 3.5 değeri için, farklı molekül sayılarında, molekül başında düşen ortalama çarpışma sayısı 10 olana kadar çalıştırılması için geçen zamanda yapılan duvar çarpışmalarını göstermektedir. Bu değer farklı hücre tipleri için aynıdır.

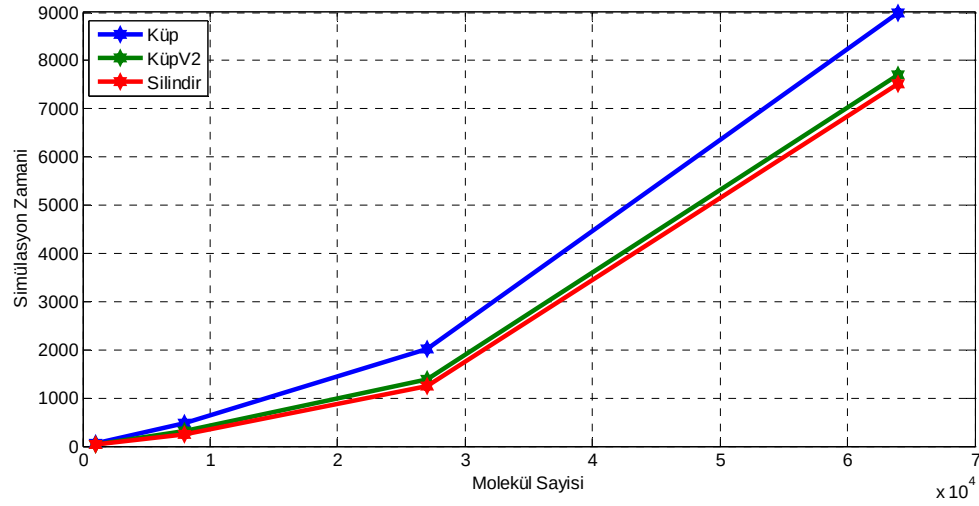


Şekil 6.3 Spacing Ratio X Duvar Çarpışması

Grafikten de görüldüğü üzere, spacing ratio ile duvar çarpışması sayısı doğru orantılıdır. Bu da demektir ki, moleküller seyrekleştikçe, belli sayıda molekül çarpışmasına ulaşmak için geçen sürede yapılan duvar çarpışması miktarı artar. Bunun yanında beklendiği üzere, ortamdaki molekül yoğunluğu aynı olsa bile molekül sayısı arttıkça duvar çarpışması miktarı artar.

6.2.4 Molekül Sayısı-Simülasyon Zamanı İlişkisi

Aşağıdaki grafikte farklı hücre tiplerinde verilen molekül sayılarında simülasyon zamanları karşılaştırılmıştır. Buna ilaveten, molekül sayısı artışının simülasyon zamanıyla ilişkisi görülebilir. Veriler spacing ratio 3.5 değerinde iken toplanmıştır.

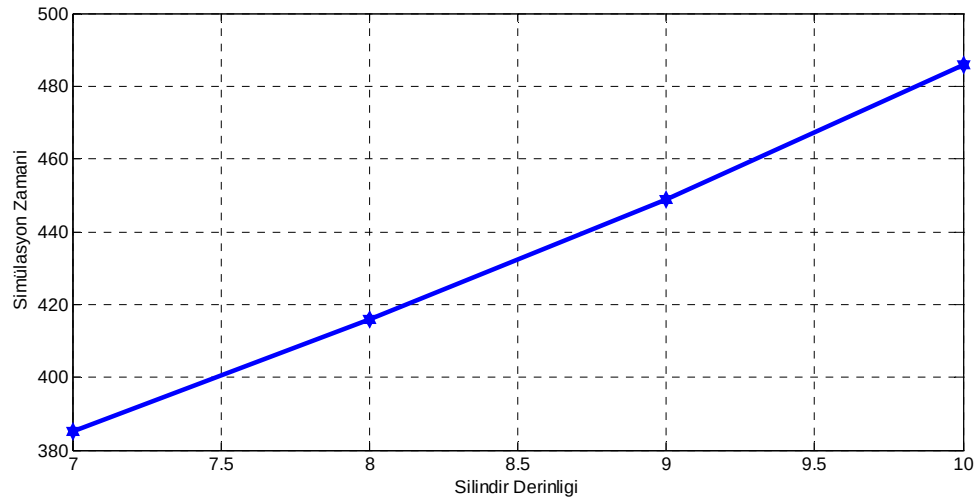


Şekil 6.4 Molekül Sayısı X Simülasyon Zamanı

Grafikte de açıkça görüldüğü üzere, silindir hücre tipi maksimum performansı sağlamıştır. Klasik hücre yaklaşımı olan küp ise en uzun simülasyon zamanına sahiptir. Bu da önerme ile tutarlıdır. Komşu hücre sayısının azalması simülasyon performansını arttırır.

6.2.5 Silindir Hücre Tipi İçin Derinlik-Simülasyon Zamanı İlişkisi

Silindir hücre tipinde, spacing ratio 3,5 değeri ve 8.000 molekül, 80.000 molekül çarpışması için, silindir yarı çapı sabit tutulup derinlik arttırıldığında aşağıdaki grafik elde edilmiştir.



Şekil 6.5 Silindir Derinliği X Simülasyon Zamanı

Grafikten de görüldüğü üzere, silindir derinliği arttıkça, simülasyon zamanı da uzamıştır. Bunun sebebi ise hücre sayısının azalması ve dolayısıyla bir hüreye düşen ortalama molekül sayısının artmasıdır.

7 SONUÇLAR

Üç hücre tipi için alınan sonuçlar değerlendirildiğinde K pV2'nin K p'e g re, silindirine ise K pV2ye g re daha performanslı olduėu g zlenmiřtir. Bu da bizim h cre sayısının azalmasının sim lasyon performansını arttıracadı tezimizi doėrulamaktadır. Sim lasyon hızlarındaki artış h cre sayısının azalmasıyla yaklaşık olarak doėrusaldır. Ancak molek l sayısı artık a sistem performansındaki artış beklenene g re azalır. Bu sapmanın kaynaėı ise sırasıyla řunlar olabilir:

- K pV2de K p'e g re h cre ge iř sayısının artması.
- Silindirdeki h cre ge iřlerini hesaplamak i in 2.dereceden bir denkleme ihtiya  duyulması: Bu sebeple, molek l sayısı  ok arttırıldıėında performans artışı olduk a fazla azalmıřtır.
- Kodlama sırasında yapılan performansı d ř r c  etmenler.  rneėin heap yerine sıralı liste kullanılması. Molek l sayısı  ok arttıėında olayları sıraya koymak ve listeden sıralı olmayarak bir elemanı se mek doėrusal zaman alır. Bu da  arpıřma hesabı zamanına g re daha baskın sonuçlar doėurabilir ve sitemin t m  d ř n ld ė nde performans artışı olmasına raėmen bariz bir řekilde g zlemlenemeyebilir.

Sim lasyon kodunda yapılacak bir optimizasyon ile, y ksek sayıda molek ller i in de performans artışını doėrusala yaklařtırılabilir. İleriki  alıřmalarda bu optimizasyon yapılacak ve daha farklı h cre tipleri uygulanacaktır. Yapılan deneyler sonucunda yukarıda sayılan fakt rlerin etkisinin minimum olduėu -molek l sayısının 100.000den k   k olduėu durumlarda- beklenen performans artışı saėlanmıřtır. Sonu  olarak, bu  alıřma sayesinde  ok h creli bir molek l dinamiėi sim lasyonunda, bir h credeki ortalama molek l sayısı sabit tutularak, komřuluk sayısının azaltılmasının sistemin performansını arttıracadı g r lm řtir.

8 ÖNERİLER

İleriki çalışmalarda iç içe geçmiş hücre yapıları kullanılarak komşuluk sayısı daha fazla azaltılabilir. Farklı hücre yapılarıyla, olay temelli bir molekül dinamiği simülasyonu yapıldıktan sonra, farklı sınır koşullarında simülasyon tekrar edilebilir. Bu sayede molekül davranışları ile ilgili daha detaylı bilgi edinilebilir. Ayrıca simülasyon ortamının istenilen bir kısmının kesitler halinde görüntüleri alınabilir. Bu görüntüler sayesinde o kesitin basıncı, molekül yoğunluğu ya da sıcaklığı gibi konularda bilgi edinilebilir. Ayrıca daha sonraki aşamalarda moleküllerin tepkimeye girmesi gibi durumlar da simülasyona eklenebilir. Son olarak ise geliştirilen simülasyon performans artışını yitirmeden paralelleştirilerek aynı anda gözlemelenebilen (simülasyona katılan) molekül sayısının artırılması sağlanabilir.

KAYNAKLAR DİZİNİ

- [1] İlyas KANDEMİR, A Multicell Molecular Dynamics Method, Case Western Reserve University, Source DAI-B 60/03, Sf. 1154, (1999)
- [2] 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)
- [3] 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)
- [4] 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)
- [5] 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)
- [6] S. Miller¹ ve S. Luding¹, Event-driven molecular dynamics in parallel, J. of Comp. Phys., Kısım 193, Nüsha 1, (2004)
- [7] Rapaport D., The Event Scheduling Problem in Molecular Dynamic Simulation, J. of Comp. Phys., Kısım 34, Sf.184- 201, (1980)
- [8] Boris D. Lubachevsky, How to Simulate Billiards and Similar Systems, J. of Comp. Phys., Kısım 94, No 2, Sf. 255-283, (1991)

9 ÖZGEÇMİŞ

1981 yılında Bursa’da doğdu. Orta ve Lise Eğitimini Bursa Cumhuriyet Lisesinde tamamladıktan sonra, ODTÜ Bilgisayar Mühendisliği Bölümünü bitirdi. Meteksan Sistem AŞ. de yazılım mühendisi olarak ve TÜBİTAK Marmara Araştırma Merkezinde araştırmacı ünvanıyla çalıştı. Halen TÜBİTAK-MAM bünyesindeki Bilişim Teknolojileri Enstitüsünde çalışmaktadır.