



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

**MOLEKÜLER MODELLEME METODU İLE
NESNELERİN HACİMSSEL OLARAK MODELLENMESİ
VE DEFORMASYONU**

Rıdvan YAYLA
Yüksek Lisans Tezi

Tez Danışmanı
Yrd. Doç. Dr. Alpaslan DUYSAK

BİLECİK, 2012

 BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI	YÜKSEK LİSANS JÜRİ ONAY FORMU
---	--

Bilecik Şeyh Edebâli Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nuntarih ve sayılı kararıyla oluşturulan jüri tarafından tarihinde tez savunma sınavı yapılan Rıdvan YAYLA'nın "Moleküler Modelleme Metodu ile Nesnelerin Hacimsel Olarak Modellenmesi ve Deformasyonu" başlıklı tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak oy birliği/oy çokluğu ile kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Yrd. Doç. Dr. Alpaslan DUYSAK

ÜYE : Yrd. Doç. Dr. Doğan AYDIN

ÜYE : Yrd. Doç. Dr. Ü. Çiğdem TURHAL

ONAY

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun/...../.....tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

ÖZET

Nesnelerin gerçek zamanlı ve fiziksel olarak doğru bir şekilde deformasyonlarının elde edilebilmesi için çeşitli metotlar ve algoritmalar geliştirilmiştir. Simülasyon uygulamaları yaygın olarak Kütle-Yay Sistemleri ve Sonlu Elemanlar Metodu kullanılarak gerçekleştirilir. İnsan organları, animasyon karakterleri gibi deforme edilecek nesneler yüzeysel olarak veya hacimsel olarak modellenir. Yüzeysel modellemede çoğunlukla üçgenler kullanılır. Bu modelleme metodu hacim bilgisini dikkate almaz ve sonuçları da gerçekçi değildir. Hacim bilgisini hesaba katan modelleme tekniklerinde genellikle üçgen prizma (tetrahedral) geometrik yapı elemanları kullanılır.

Moleküler modelleme teknolojisi hızlı çalışan ve gerçekçi iç kuvvetler üreten bir metottur ancak yüzeysel modelleme için geliştirilmiştir. Bu tezde moleküler modelleme algoritması hacimsel olarak modellenmiş nesnelerin deformasyonunda kullanılmak üzere modifiye edilmiş, yeni kuvvet fonksiyonları elde edilmiş ve deformasyon algoritmasında kullanılmıştır.

Yeni algoritma değişik nesnelerin deformasyonunda kullanılmış ve göze hoş gelen simülasyonlar ve gerçeğe daha yakın sonuçlar elde edilmiştir. Kütle-yay sistemi ve yüzey modelleme ile karşılaştırıldığında hız konusunda taviz verilmesine karşın birçok model için gerçek zamanlı simülasyonlar yapılabilmektedir.

Anahtar Kelimeler

Deformasyon, Kütle-Yay Sistemi, Moleküler Modelleme ,Tetrahedral Model

ABSTRACT

Various methods and algorithms are developed because deformations of objects are obtained that being correct in real time as physical. Simulation applications widely are realized by using the Mass-Spring System and Finite Element Method. The objects that will be deformed are modeled as surface or volume such as human organs, animation characters. Triangles are generally used for surface modeling. This modeling method doesn't consider the volume information and its results don't realistic. Tetrahedral geometric shape is widely used in the modeling methods that take account to volume information.

Molecular modeling technology is a method that rapidly works and produce realistic internal forces but it is developed for surface modeling. In this thesis, molecular modeling algorithms is modified for use in deformations of objects that modeling as volume, new force functions are obtained, they are used for deformation algorithm.

New algorithm is used for deformation of different objects and simulation that pleasing to the eye to come and the results that are more recently to real are obtained . While the new algorithm is compared mass-spring system and surface modeling, although it is made a concession about speed, simulations in real time can be made .

Key Words

Deformation, Mass-Spring System, Molecular Modeling ,Tetrahedral Model

TEŞEKKÜR

Tez çalışmam boyunca bilgisini ve desteğini benden hiçbir zaman esirgemeyen değerli danışmanım Yrd. Doç. Dr. Alpaslan DUYSAK'a, Bilecik Şeyh Edebali Üniversitesi Bilgisayar Mühendisliği bölüm hocalarıma, motivasyon ortamını her zaman yüreğimde hissettiğim değerli iş arkadaşlarıma, maddi ve manevi desteğini hiç bir zaman benden esirgemeyen değerli aileme teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET.....	iii
ABSTRACT.....	iv
TEŞEKKÜR.....	v
İÇİNDEKİLER.....	vi
ÇİZELGELER DİZİNİ.....	ix
ŞEKİLLER DİZİNİ.....	x
SİMGELER VE KISALTMALAR DİZİNİ.....	xii
1. GİRİŞ.....	1
1.1. Simülasyon	1
1.2. Simülasyonun Kullanım Alanları.....	2
1.2.1. Eğitim & Oyun Programlama.....	2
1.2.2. Tıp.....	3
1.2.3. Askeri Alan.....	4
1.2.4. Sanayi.....	4
1.3. Sanal Gerçeklik (SG).....	5
1.4. Sanal Ortam.....	6
2. LİTERATÜR ÇALIŞMASI.....	7
2.1. Kütle-Yay Sistemi Alanında Yapılan Çalışmalar.....	7
2.2. Tetrahedral Modelleme Alanında Yapılan Çalışmalar.....	7
2.3. Sonlu Elemanlar Metodu ile Yapılan Çalışmalar.....	8
3. KÜTLE-YAY SİSTEMİ.....	11
3.1 Deformasyon Nedir?.....	11
3.1.1 Fiziksel Yapı.....	11
3.1.2 Geometrik Şekil.....	12
3.2. Hooke Yasası ve Yaylar.....	13
3.3. Kütle-Yay Sisteminde Yaylar.....	16
3.4. Euler ve Verlet Yöntemi.....	17

4.	MOLEKÜLER MODELLEME.....	18
4.1.	Yüzey Moleküler Modelleme.....	18
4.1.1.	Yüzey Üçgen Modelleme.....	18
4.1.1.1.	Yüzeysel Olarak Modellenmiş Sistemlerde Moleküler Modelleme Metodu.....	18
4.1.1.2.	Yüzey-Açı Potansiyel Enerjisi.....	19
4.1.1.3.	Bağ-Açı Potansiyel Enerjisi.....	19
4.1.1.4.	Gerilme-Sönümlleme Sonucu Oluşan Potansiyel Enerjisi.....	20
4.1.1.5.	Toplam İç Kuvvet ve Sistem Dinamiği.....	20
4.2.	Hacimsel Moleküler Modelleme.....	22
4.2.1.	Tetrahedral Modelleme.....	22
4.2.1.1.	Tetrahedral Modellemede Kütle-Yay Sistemi.....	24
4.2.1.1.1.	Tetrahedral Modellemede Yüzey Kuvvetleri...	26
4.2.1.1.2.	Tetrahedral Modellemede Bağ-açı Kuvvetleri.	26
4.2.2.	Hacimsel Kuvvet Hesabı.....	28
4.2.2.1.	Biss-Was Haman Potansiyel Enerji Fonksiyonu.....	28
4.2.2.2.	Biss-Wass Haman Potansiyel Enerji Fonksiyonu ile Elde Edilen Kuvvetler.....	28
4.2.2.3.	Khor-Dass Potansiyel Enerji Fonksiyonu.....	31
4.2.2.4.	Khor Dass Potansiyel Enerji Fonksiyonu ile Elde Edilen Kuvvetler.....	32
5.	DEFORMASYON SİMÜLASYONU.....	33
5.1.	OpenGL.....	33
5.2.	Geometrik Yapı.....	34
5.3.	Modelleme.....	35
5.4.	Uygulama Adımları.....	36
5.5.	Nesnelerin Sanal Ortama Aktarılması.....	36
5.6.	Ağ Oluşumu (Meshing).....	37

5.7.	Normallerin Hesaplanması.....	37
5.8.	Modellenen Nesnede Deformasyon.....	40
5.9.	Khor-Dass ve Biss-Wass Haman Potansiyel Fonksiyonu ile Elde Edilen Kuvvetlere Euler ve Verlet Metodlarının Uygulanması.....	41
5.10.	Moleküler Modelleme ve Hacimsel Modelleme ile Yapılan Deformasyonun Hız Testleri.....	43
6.	SONUÇ VE ÖNERİLER.....	46
7.	EKLER.....	48
	KAYNAKLAR.....	56
	ÖZGEÇMİŞ.....	60

Sayfa No

TEZ ONAY SAYFASI

ÇİZELGELER DİZİNİ

Çizelge 3.1. Karbon, Silisyum ve Germanyum için Khor-Dass potansiyel fonksiyonu parametreleri.....	38
---	----

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 1.1:	Eğitimsel amaçlı kullanılan bir sürüş simülatörü	3
Şekil 1.2:	Bir simülatör odasındaki tıbbi simülatör cihazı	3
Şekil 1.3:	Askeri bir eğitimde kullanılan helikopter simülatörünün kokpit.....	4
Şekil 1.4:	Simülasyon uygulamasında kullanılan bir Haptic kol.....	5
Şekil 3.1:	Kütle yay sistemine ait nesne modeli.....	12
Şekil 3.2:	(a) Triangle geometrik şekil ile oluşturulan içi boş insan gövdesi modeli (b) Tetrahedral geometrik şekil ile oluşturulan içi dolu insan gövdesi modeli.....	13
Şekil 3.3:	Triangle model ile yapılmış Standford tavşanı.....	13
Şekil 3.4:	Triangle model ile yapılmış Happy Budha heykeli.....	13
Şekil 3.5:	Hooke yasasına göre yay gerilmesi.....	14
Şekil 3.6:	Bir tetrahedral modele ait Kütle-Yay Sistemi.....	15
Şekil 3.7:	Tetrahedral şekil ile modellenmiş mide modeli (solda) Deformasyona uğramış bir mide modeli (sağda).....	16
Şekil 4.1:	Atomlar arası bağlara ilişkin örnek bir gösterim.....	18
Şekil 4.2:	Deformasyon öncesi denge durumunda olan kütle yay modeli.....	19
Şekil 4.3:	Deformasyon ile birlikte oluşan bağ-açı potansiyel enerjisine ait kütle yay modeli.....	19
Şekil 4.4:	Deformasyonda gerilme-sönümlemeye ait kütle-yay modeli	19
Şekil 4.5:	Tetrahedral Geometrik Şekil.....	22
Şekil 4.6:	Düzgün dört yüzlü (yüzeyler).....	23
Şekil 4.7:	Tetrahedral modelleme ile oluşturulmuş bir limon modeli.....	23
Şekil 4.8:	Tetrahedral model ile oluşturulmuş bir mide modelinin ilk deformasyona uğramış hali	23
Şekil 4.9:	Tetrahedron yüzeylerine ait gösterim.....	24
Şekil 4.10:	1.üçgen yüzeye ait gösterim.....	24
Şekil 4.11:	2.üçgen yüzeye ait gösterim.....	25

Şekil 4.12:	3.üçgen yüzeye ait gösterim.....	25
Şekil 4.13:	4.üçgen yüzeye ait gösterim.....	26
Şekil 4.14:	Tetrahedral modellemede üçgen yüzeylere ait deformasyon sonucu oluşan şekle bağlı bağ açısı kuvvetleri değerleri.....	27
Şekil 5.1:	OpenGL kütüphanesine ait glut32.dll dosyasının eklenmesi.....	33
Şekil 5.2:	OpenGL kütüphanesine ait glut32.lib dosyasının eklenmesi.....	33
Şekil 5.3:	OpenGL kütüphanesine ait glut.h dosyasının eklenmesi.....	34
Şekil 5.4:	OpenGL kütüphanesinin C++ platformuna aktarılması.....	34
Şekil 5.5:	OpenGL dili ile yapılmış bir tetrahedral model.....	34
Şekil 5.6:	Tetrahedral Geometrik Şekil.....	34
Şekil 5.7:	Tetrahedral modelin nokta bilgilerinin tutulduğu tetgen dosyası.....	35
Şekil 5.8:	Tetrahedral modele ait noktaların koordinatların tutulduğu dosya.....	35
Şekil 5.9:	Tetrahedral Modellemede bir modele ait uygulama adımları.....	36
Şekil 5.10:	Tetrahedral Geometrik Şekillerin Tutulduğu Struct Yapısı.....	36
Şekil 5.11:	Tetrahedron ağ oluşumunun temel bir görüntüsü.....	37
Şekil 5.12:	Bir üçgenin noktaları.....	37
Şekil 5.13:	Normal ortalaması alınmadan modellenmiş böbrek modeli.....	39
Şekil 5.14:	Normal ortalaması alınarak modellenmiş böbrek modeli.....	39
Şekil 5.15:	Yüzey kaplaması ve tetrahedral modele ait üçgen çizimine ait kod bloğu.....	39
Şekil 5.16:	Deformasyon sonucu bağlantılı noktaya etki eden yer değiştirmelere ait kod bloğu.....	40
Şekil 5.17:	Bir tetrahedral geometrik şekle bağlı olan uçları kontrol eden kod bloğu.....	41
Şekil 5.18:	Biss-wass haman ve Khor-dass kuvvetleri ile oluşturulmuş bir deformasyon örneği.....	41
Şekil 5.19:	(a) Biss-wass Haman kuvvetlerinin hesaplanmasına yönelik programın 1. blok parçası.....	42
	(b) Biss-wass Haman kuvvetlerinin hesaplanmasına yönelik programın 2. blok parçası.....	43
Şekil 5.20:	Khor-dass kuvvetlerinin hesaplanmasına yönelik program bloğu.....	43

Sayfa No

Şekil 5.21:	Hız testinde kullanılan karaciğer modeli.....	44
Şekil 5.22:	Moleküler modelleme ile yapılan deformasyonda geçen süre.....	44
Şekil 5.23:	Khor-dass ve Biss-wass Haman kuvvetleri ile yapılan deformasyonda geçen süre.....	44
Şekil 5.24:	Hız testinde moleküler modelleme ve hacimsel modellemede geçen süreyi gösteren grafik.....	45

KISALTMALAR DİZİNİ

Simgeler

Açıklama

3B	:	3 Boyut
bmp	:	Windows bitmap resim formatı
KYS	:	Kütle-Yay Sistemi
ms	:	milisaniye
PEF	:	Potansiyel Enerji Fonksiyonu
SG	:	Sanal Gerçeklik
SEM	:	Sonlu Elemanlar Metodu

1. GİRİŞ

3 Boyutlu (3B) Modelleme ve simülasyonun ana hedefi, nesnelerin asıllarına uygun olarak sanal ortama aktarılması ve gerçek zamanlı olarak görüntülenmesidir. Bilgisayar grafikleri teknolojisinin hızla yayıldığı günümüz dünyasında nesnelerin simülasyonunun oluşturulması için geometrik ve fiziksel şekillerden yararlanılarak bir çok teknik geliştirilmiştir. Bu teknikler sayesinde bilgisayar grafik tasarımı birçok uygulama alanına olanak vermiştir; Cerrahi simülatör araçları, bilgisayar oyunları ve animasyon filmleri bunlardan bazılarıdır.

Özellikle tıp alanında çalışacak olan uzman hekimlerin cerrahi eğitimlerinde deneyim kazanmaları için kullanılmaktadır. Ameliyatların gerçekleşmesinden önce yapılan cerrahi eğitimlerin gerçekleştirilmesi için simülatör araçlarından yararlanılmaktadır. Simülatör araçları sayesinde hekim adayları, ameliyat sırasında nasıl hareket edeceklerini, organ modelleri üzerinden nasıl operasyon yapacaklarını uygulamalı olarak deneyebileceklerdir.

Sanayi alanında , makine üretiminde kullanılan cihazların model gösterimleri için modelleme tekniklerinden yararlanılmaktadır. Savunma sanayisinde belirli bir alanın simüle edilebilmesi için, 3B modellemeden yararlanılmaktadır.

Modellemeler genellikle üçgen veya tetrahedral yapı türleri kullanılarak yapılmaktadır. Üçgensel (triangle) modeller üzerinden yola çıkılarak, tetrahedral model geliştirilmiştir. Bu tezde, nesnelerin modellenmesi tetrahedral yapılar ile gerçekleştirilmiş ve kütle-yay sistemi ve moleküler modelleme yöntemleri temelinde deformasyon simülasyonu gerçekleştirilmiştir.

1. Simülasyon

Simülasyon, diğer bir adıyla benzetim olarak literatüre geçmiş olan bu yöntem, Akgül ve arkadaşlarına göre mevcut olan teorik ya da fiziksel bir sistemin bilgisayar ortamında modellendikten sonra farklı koşullar altında vereceği sonuçları gerçek sistemle karşılaştırma, alternatif senaryolar geliştirerek üretimde mükemmelliği yakalayabilme imkanı olarak tanımlanmaktadır. (Akgül M.K., Aydın C., Çarkıt N, vd 2006). Simülasyon teknikleri sayesinde bir nesnenin bilgisayar ortamında modellenerek, modellenen nesneye ait fiziksel özellikleri araştırılabilmekte, elde edilen sonuçlar sayesinde fiziksel ortamdan etkilenen nesnenin yeniden

modellenmesi sağlanabilmektedir. Bu kapsamda yapılan çalışmalar ile birçok sanal araştırmaya zemin hazırlanmaktadır.

1.1. Simülasyonun Kullanım Alanları

Simülasyon son yıllarda eğitim alanından, tıbbi cihaz programlamasında, savunma sanayisinde askeri eğitimlerden, oyun programlamaya, büyük sanayi bölgelerinde makine programlamasına kadar bir çok alanda gelişim göstermiştir. Bu alanlarda kullanılan simülatörler kullanım amacına bağlı olarak, kullanıcılarına birçok değişik imkan sunmakta ve iş yaşamındaki işleyişi kolaylaştıran bir unsur olmaktadır. Bazı eğitimlerin görsel olarak adaylara sunulması ile yapılan çalışmalar, simülasyon programlarının ve cihazlarının kullanılması ile gerçek yaşamdaki hataları en aza indirmektedir. Simülasyon araçlarının eğitimlerde kullanılması, kullanıcıların kendi alanlarındaki eğitimin kolaylaştırılmasını sağlamaktadır. Bu sayede gerçek bir eğitim ortamında eğitim alan kullanıcılar, simülasyon araçları sayesinde gerçek bir ortamda oluşabilecek durumlara karşı önceden fikir sahibi olmaktadır. Simülasyon cihazları gerçek yaşamda karşılaşılabilecek muhtemel durumlara uygun olarak yapılandırılmış cihazlardır. Bu alanlarda yapılan çalışmalar modelleme tekniklerinin fiziksel boyutunun önemini de ortaya koymaktadır. Ayrıca modern dünyada şehirleşmenin hızla yaygınlaştığı yerlerde, şehir ve bölge planlama alanında da simülasyon ortamlarından yararlanılmakta, yerleşim birimlerinin bu sayede düzenli olarak yaygınlaştırılması amaçlanmaktadır. Simülasyonun ön plana çıktığı kullanım alanları şu şekilde sıralanmaktadır.

- Eğitim & Oyun Programlama
- Tıp
- Askeri Alan
- Sanayi

1.2.1. Eğitim & Oyun Programlama

Günümüz dünyasında yaygınlaşan oyunlar, simülasyon alanında da kendini göstermeye başlamıştır. Özellikle bilgisayar grafik teknolojisi ile gerçeğe yakın görüntünün elde edilmesi sayesinde oyunlarda, kullanıcıların gerçek ortamda yaptıkları oyun hareketlerinin bilgisayar ortamına taşınmasını sağlamıştır. Sanal ortamda gerçekleştirilen oyunlar, özellikle eğitim alanında da öğrencilerin psikolojik ve bilişsel açıdan gelişmeleri için eğitici oyunlar olarak ön plana çıkmıştır. Ayrıca araç

kullanımına teşvik etmek amacıyla kullanıcıların gerçek ortamda karşılaşılabilecekleri manevraları göstermek amacıyla Şekil 1.1’de gösterildiği gibi araç simülatörlerinden yararlanmaktadır. (Scott H., Knowles M., vd. , 2012)



Şekil 1.1. Eğitimsel amaçlı kullanılan bir sürüş simülatörü. (Scott H., Knowles M., vd. , 2012)

1.2.2. Tıp

Çeşitli meslek dallarında gelişimini sürdüren simülatör cihazları özellikle tıp alanında doktor adaylarına çeşitli operasyon denemeleri öncesinde deneyim kazanmalarını sağlamak amacıyla geliştirilmiştir. (Scalese Ross J. , Obeso Vivian T. , Issenberg S. Barry , 2008) Cerrahi simülatörlerde kullanılan mankenler ile gerçek ortam, doktor adaylarına sunulmakta ve muhtemel senaryolara karşı doktorların eğitimine imkan sağlamaktadır. Simülatörler gerçek bir ortama entegre edilerek oluşturulmakta ve bu amaçla gelişmiş simülasyon merkezleri yapılandırılmaktadır. Ayrıca simülasyon ekranı sayesinde müdahale edilen noktaya olan temasları güç geri besleme ile cerrah adayı hisseder. Şekil 1.2 ’de bu yapılandırma ortamında oluşturulmuş olan simülasyon ile yapılan bir operasyon örneği gösterilmiştir. (Mendoza C., vd. 2002)



Şekil 1.2. Bir simülatör odasındaki tıbbi simülatör cihazı. (Mendoza C., vd. 2002)

1.2.3. Askeri Alan

Küresel dünyada savunma sanayisinde yaşanan günlük gelişmeler, askeri alanda farklı tekniklerin oluşturulmasına imkan sağlamıştır. Askeri alanda özellikle hava savunmasına ait olan anlık değişimler simülatör cihazları sayesinde kullanıcılara görsel bir bilgi sunmakta ve alınacak önlemlere karşı yetkili organları bilgilendirmektedir.



Şekil 1.3. Askeri bir eğitimde kullanılan helikopter simülatörünün kokpiti. (Rehmeiner M., 2008)

Ayrıca askeri operasyonlarda bir bölgenin gerçeğe yakın bir görüntüsünün elde edilerek simülatör cihazlarına aktarılması, bölgeye ait alınacak önlemler ve savunma taktikleri konularında askeri yetkilileri bilgilendirmek amacıyla oluşturulmuştur. Eğitsel olarak askeri ortamlarda pilot yetiştirilmesi amacıyla oluşturulan helikopter ve uçak simülatörleri, pilot adaylarının gerçek bir hava aracını kullanmadan önce deneyim kazanmaları için büyük bir önem arz etmektedir. Şekil 1.3'de bu amaçla oluşturulan bir helikopter simülatörü gösterilmiştir. (Rehmeiner M., 2008)

1.2.4. Sanayi

Simülasyon tekniklerinin yaygınlaşmasıyla sanayi alanında da birçok yenilik ortaya konulmuş ve makine üretiminde simülatör cihazlarının etkin rolü ön plana çıkmıştır. Otomotiv sanayisinde araçların daha nitelikli olarak üretilmesi için oluşturulan simülasyon ortamları, daha kullanışlı ve daha az yakıt ile daha fazla mesafe almayı amaçlayan tasarımların oluşturulmasını sağlamaktadır (Morris A., Kok D., vd. 2012). Sanayi-üniversite işbirliği ile oluşturulan Ar-ge projelerinde simülasyon ortamlarına ihtiyaç duyulmakta, üretilecek bir makinenin üretimden önce daha kullanışlı olarak üretilmesine imkan sağlamaktadır (Odabaşı Y., Helvacıoğlu Ş., vd. 2010).

Bu sayede zaman ve üretim tasarrufu sağlanmakta ve işveren kesimine daha uygun imkanlar tanınmaktadır. Niteliksel olarak ifade edilen simülasyon araçları sanayi alanında daha az hata ve daha az maliyet ile üretimin hız kazanmasına olanak tanımaktadır. Bayraktar ve Kaleli (2007), dünyanın en ağır iş makineleri üreticilerinden biri olan Caterpillar Inc., Peoria, IL., tasarımları gözden geçirmede pahalı ve zaman alıcı olan gerçek prototipler yerine sanal prototipleri tercih ederek, tercih, fiyat ve zaman esaslı rekabette yeni ürünleri pazara en kısa süre ve en uygun fiyatlarla getirdiklerini belirtmektedir.

1.1. Sanal Gerçeklik (SG)

Sanal gerçeklik (SG) kavramı, insana gerçek hissi veren bilgisayar grafik teknolojisi aracılığıyla gerçek ortamda bulunan bir nesnenin bilgisayar ortamındaki etkilerinin araştırılmasına yönelik bir kavram olarak ortaya çıkmıştır. Bilgisayar grafik teknolojilerinin hızla gelişimi ile nesnelerin gerçek zamanda modellenmesi ve gerçeğe yakın görüntüsünün elde edilmesiyle sanal gerçeklik kavramı önem kazanmıştır. Sanal gerçeklik, bir nesneye gerçek ortamda etki eden her türlü etkinin sanal ortamda gerçek zamandaki hızına yakın bir şekilde simüle edilmesini ifade eder. Sanal gerçeklik gerçek bir ortamda kullanıcının sahip olabileceği kontrol hissinin bütünüyle kullanıcının kendisine verilmesini de ifade eder.



Şekil 1.4. Simülasyon uygulamasında kullanılan Haptic kol. (Aytekin S. 2012)

Kullanılan cihazlar ile kontrol hissinin veren gözlük, eldiven, haptic kol, oturma paneli gibi özel yapıma bağlı olan aletler geliştirilmiştir. Şekil 1.4’de dış kuvvete karşı geri bildirimi sağlayan duyarlı Haptic Kol gösterilmektedir. (Aytekin S. 2012)

Sanal gerçeklik çerçevesinde incelenen ve sanal gerçekliğin uygulama alanları ile ilişkilendirilen çalışmalar, sanal prototipler ile mühendislik analizlerine katkıda bulunmuştur. Bayraktar ve Kaleli (2007) , sanal gerçekliğin e-ticareten, sanayi alanına

eğitimden, genel sanal gerçeklik uygulamalarına kadar birçok alandaki çalışmayı incelemiştir. (Bayraktar K., Kaleli F., 2007)

1.2. Sanal Ortam

Sanal ortam, genel anlamda bir nesneye bağlı olarak gerçekleştirilen 3 boyutlu bir nesne modelinin bilgisayar grafikleri yardımıyla bilgisayar ortamına aktarıldığı bir platformdur. 1945’li yılların başlarında gelişim gösteren bilgisayar teknolojisi, sanal ortamlar sayesinde belli hesaplamaların yapılmasını sağlamıştır. Bilgisayar grafik teknolojisi ile geliştirilen sanal ortamlar, nesne modelinin ve bu modele etki eden dış kuvvetlerin model üzerindeki etkilerinin görüntülenmesini sağlamıştır.

Nesnelerin sanal ortamda görüntülenmesi ve bu ortamda nesnede oluşan deformasyonların görüntülenmesi için çeşitli yazılımlar geliştirilmiştir. Bu kapsamda simülasyon ve modellemeye yönelik olarak kullanılan programlar ve derleyiciler şu şekilde sıralanabilir:

- 3D Max Studio, Maya vb. paket programlar,
- C, C++ , Java,
- DirectX,
- OpenGL,

Yukarıda sıralanan programlar ve derleyiciler farklı ihtiyaçlara cevap vermektedir ve genellikle birbirlerini tamamlarlar ve bu kapsamda birlikte kullanılırlar.

2. LİTERATÜR ÇALIŞMASI

2.1. Kütle-Yay Sistemi Alanında Yapılan Çalışmalar

Kütle-yay sistemi, modelleme alanında birçok teknik ile birlikte kullanılan bir yöntemdir. Kütle-yay modeline ilişkin yapılan çalışmalar sanal gerçeklik bünyesinde deformasyon simülasyonlarında kullanılmaktadır. Bu kapsamda Mollemans ve arkadaşları (2003) çene cerrahisi için tetrahedral kütle-yay modeli ile yumuşak doku deformasyonu için yeni bir yöntem geliştirmiştir(Mollemans W., Schutyser F., Cleynenbreugel J.V., vd. 2003). Xu ve diğerleri ise (2007) yumuşak doku deformasyonu ile gerçek zamanlı olarak nesnelerin fiziksel kıyaslamasını yapmış, sezgisel optimizasyon tekniklerinden yola çıkarak kütle-yay modelinin deforme edilebilir nesneler üzerindeki etkilerini analiz ederek, gerçek zamanlı incelemelerde bulunmuştur (Xu J., Zhang J., Huang P., vd. 2007). Kütle-yay sistemi temelinde yapılan çalışmalar yalnızca medikal alanda değil, farklı sanal gerçeklik alanlarında da yapılmaktadır. Bu bağlamda Zhendong (2011) çalışmalarında gerçek zamanlı kumaş deformasyonunu kütle-yay sistemi temelinde incelemiş, katı cisimlerin hacimsel modelleme yöntemi ile deformasyonunu analiz etmiştir. (Zhendong L., 2011)

2.2. Tetrahedral Modelleme Alanında Yapılan Çalışmalar

Tetrahedral Modelleme, günümüz dünyasında sanal gerçeklik alanında kullanılan yaygın bir tekniktir. Tetrahedral modelleme, tetrahedral geometrik şekil temelinde nesnelerin hacimsel olarak modellenmesini sağlayan bir modelleme yöntemidir. Nesnelerin yapıldığı maddelerden yola çıkılarak, esnekliklerinin dikkate alınması ve bu sayede disiplinler arası bir çalışmanın ortaya konulduğu bir yöntemdir. Özellikle biyomedikal alanda kullanımı yaygın olan simülasyon teknikleri, farklı modelleme tekniklerinin gelişmesi ile doku ve organların sanal ortamlara aktarılmasında öncü rol oynamaktadır. Delingette (1998), çalışmasında özellikle medikal simülasyonlarda, bilgisayar grafiklerinin kullanımına ilişkin, geometrik şekillerin kullanımını baz alarak analizler yapmış, gerçek doku deformasyonuna yönelik karşılaştırmalara yer vermiştir . (Delingette H., 1998)

Wang ve arkadaşları (2003), 3 boyutlu haritalama yöntemi ile tetrahedral modellemeyi birleştirerek beyin verileri ile sanal ortamda beyin modelinin görüntülenmesini başarmıştır (Liang X., Wang W., vd. 2003). Fang ve arkadaşları (2009), siyah beyaz görüntü ve hacimsel binary kodları ile tetrahedral ağ görüntüsünü

elde etmeyi başarmıştır (Fang Q., Boas D.A., 2009). Aynı şekilde Zhang ve arkadaşları (2009), akışkan yapılarda etkileşim analizlerini kullanarak medikal görüntü verilerinden yüksek kaliteli tetrahedral ağ görüntüsünü elde etmiştir (Zhang Y. vd, 2009).

Cutler ve arkadaşları da (2004), simülasyonlarda tetrahedral modellerin görüntülenmesi için daha az tetrahedral geometrik şekil ile tetrahedral modellerin sadeleştirilmesini ve yeniden düzenlenmesini sağlamıştır (Cutler B., Dorsey J., McMillan L. vd., 2004). Weichert ve diğerleri (2010), deforme edilebilir bir yüzey modelini kullanarak yumuşak doku deformasyonu için sonlu elemanlar metodu ile yeni bir tetrahedral ağ yaklaşımı geliştirmiştir (Weichert F., Schröder A., Landes C., vd. 2010).

Tetrahedral modellemede, basitleştirme ve sadeleştirme yöntemleri ile daha az tetrahedral geometrik şekiller ile daha net görüntülerin elde edilmesi sağlanmıştır. Bu kapsamda Trotts ve diğerleri (1999), üçgensel modelleme yöntemlerini kullanarak tetrahedral modellemede veri kümelerini incelemiş ve her bir noktanın belirgin hata durumuna kadar modelin geometrisini sadeleştirerek etkin bir görsel ara yüz elde edilmesini sağlamıştır. Tahmini hataların elde edilerek, en az hata ile sadeleştirilmesi tetrahedral modellemenin kullanılması ve yaygınlaştırılması noktasında farklı yaklaşımların oluşmasını sağlamıştır (Trotts I. J., Hamann B., vd. 1999).

2.3 Sonlu Elemanlar Metodu ile Yapılan Çalışmalar

İlk olarak yapı analizinde kullanılan Sonlu elemanlar metoduna ilişkin ilk çalışmalar Hrennnikoff (1942) ve Mc Henry (1943) tarafından geliştirilen yarı analitik bir analiz metottur. Argyis ve Kelsey (1960) sanal bir iş prensibi olarak ifade edilen, sonlu elemanlar metodu terim olarak ilk defa Clough (1960) tarafından kullanılan bir çalışmada telaffuz edilmiştir. İki boyutlu modellemede başarılı bir şekilde kullanılmasından sonra üç boyutlu modellemede kullanılması kolay bir şekilde gerçekleştirilmiştir (Yaşar C., 2004).

Sonlu elemanlar metodu (SEM), karmaşık olan problemlerin daha basit alt problemlere ayrılarak her birinin kendi içinde çözümlenmesi ile tam çözümünün bulunduğu bir çözüm şeklidir. Sonlu elemanlar metodu simülasyonda hızın daha az önemli olduğu deformasyonlarda kullanılan bir yöntem olarak karşımıza çıkmaktadır. Bu yöntem 3 temel alanda incelenmektedir.

İlk olarak, geometrik olarak karmaşık olan bir model ele alınır ve sonlu elemanlar metodu ile belirlenen basit alt bölgelere ayrılır. İkinci olarak, her bölgedeki sürekli fonksiyonlar, cebirsel polinomların lineer kombinasyonu olarak tanımlanacağı şekilde kabul edilir. Üçüncü bir varsayım da, aranan değerlerin her eleman içinde sürekli olan tanım denklemlerinin belirli noktalardaki değerleri elde edilmesinin problemin çözümünde yeterli olmasıdır. Özellikle sanayi alanında makinelerin kullanılması ile pahalı deneylerin daha kolay incelenebilmesi amacıyla oluşturulmuş olan bir metottur. Daha çok katı modellemede öne çıkan sonlu elemanlar metodu katı mekaniği, sıvı mekaniği, biyomekanik, ısı transferi gibi alanlarda yaygın olarak kullanılmaktadır. sonlu elemanlar metodu, nesnelerin anlık durumlarını inceleyen ve tam sonuçlara ulaştıran bir yöntem olmasına karşın, kütle yay sistemlerine göre daha ayırık bir modelleme tekniği olduğu için, ayırık modellemede daha fazla kullanılan bir yöntemdir. Sonlu elemanlar metodu farklı kullanım alanları ile modellemeye uyarlanan etkili bir yöntemdir.

Modellemenin medikal ortamlarda yaygın hale gelmesi, sonlu elemanlar yöntemi ile oluşturulan modellerin hesaplamalarında etkili bir biçimde kullanılmıştır. Gülçimen ve arkadaşları (2011), insan ayağı biyomekaniğinin sonlu elemanlar yöntemi ile incelemiş ve sonlu elemanlar metodunun deneysel sonuçlar ile güvenilirliğini öne çıkarmıştır (Gülçimen B., Özcan R., Ülkü S., 2011). Gülçimen, çalışmasında bir insan ayağının farklı yapıları olan kemik, kıkırdak, yumuşak doku ve bağ gerilmesi ile deformasyon bilgilerinin zor olduğunu tespit etmiş, bu ihtiyaçtan yola çıkarak insan ayağının anatomik olarak gerçeğe uygun üç boyutlu geometrik modelini oluşturmuştur. Gülçimen'in Chen ve arkadaşlarına atıfta bulunduğu gibi (2001) literatürde gerçeği yansıtabilecek bir sonlu elemanlar ayak modelinin eksikliğinden yola çıkarak oluşturdukları üç boyutlu ayak modeli ile ayağın biyomekaniğini daha iyi anlamaya çalışmışlar, elde edilecek bilgilerin ayak kusurlarında yapılabilecek müdahalelere ışık tutacağını vurgulamışlardır.

Gülçimen'in atıfta bulunduğu Cheung ve arkadaşları (2005) MRI görüntülerini kullanarak kemik, yumuşak doku, ligament ve tendondan oluşan kapsamlı bir sistem ayağı modelleyerek ayağın normal biyomekanik davranışını simüle etmişler ve ayak içinde oluşan gerilme ve deformasyonları belirlemişlerdir. Oluşturulan modelin doğruluğunu görebilmek için basınç platformu kullanılan bu çalışmada, basınç

platformu ve sonlu elemanlar modelinden elde edilen ayak tabanına ait normal bası gerilmesi değerleri karşılaştırılarak deneysel sonuçlarla sayısal sonuçların birbirleriyle yeterince uyumlu olup olmadığı belirlenmiştir.

Sonlu elemanlar metodu kütle yay sistemlerine göre daha az nokta uçlarına ayrıldığı için fiziksel anlamda daha gerçekçi simülasyonlar yapmak mümkündür. Nesneye etki eden kuvvetler, belirlenmiş olan bir zaman dilimi içinde hacimsel modele ait kuvvetlerin birleştirilmesi ile denk kuvvet vektörlerine dönüştürülerek hesaplanması ve bu yol ile sonuca ulaşılması gerekmektedir. Sonlu elemanlar metodunda bu hesaplamalar için hızdan feragat edilmektedir.

3. KÜTLE-YAY SİSTEMİ

Nesnelerin sanal ortamda deformasyonlarının görüntülenmesi için kütle-yay sisteminden yararlanılır. Kütle-yay sisteminde cisimlerin nokta ve koordinat bilgileri ışığında bir ağ yapısı ile modellenir. Bu ağ modeline göre her bir nokta bir kütleyi temsil etmektedir. Deformasyon değişikliğine ait kuvvet hesaplamalarını kolaylaştırması kütle-yay modelinin sanal ortamlarda etkili bir biçimde kullanılmasını sağlamıştır. Esneklik nesneler için ayırt edici bir özelliktir. Bir cisme dışarıdan bir kuvvet uygulandığında cisim iç dinamiği ve yapıldığı madde sayesinde belli bir dayanıklılık gösterir. Cisme kuvvet uygulandığında iç özelliklerini korumak kaydı ile hacmini korumak için şekil değişikliğine gider. Bir cisme uygulanan kuvvet ortadan kalktığında bazıları eski haline geri döner, bazıları ise deforme edilmiş olarak kalır. Cisimler belli bir noktaya kadar esneklik gösterebilirler. Bu nokta esneklik sınırı olarak ifade edilmektedir. Dış kuvvet esneklik sınırını aşarsa yırtılma ya da kırılmalar meydana gelebilir. Kütle-yay sistemi bir cisme dışarıdan gelen bir kuvvete karşı, ağ oluşumunda kütleleri temsil eden noktaların birbiriyle dinamiksel olarak etkileşimi ile elde edilen bir sistem olarak tanımlanır.

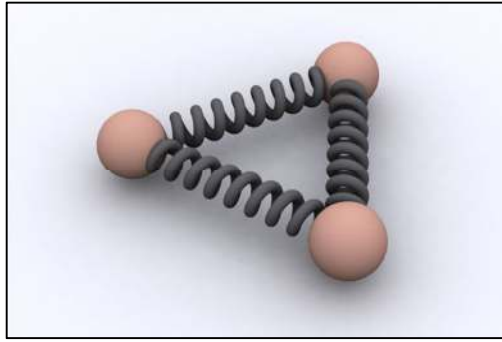
3.1. Deformasyon Nedir?

Deformasyon, bir cismin iç ve dış etkiler yoluyla itme, çekme ya da sönümlenme hareketleri ile şekil bozukluğuna uğraması, hacminin bu kuvvetler yoluyla değişiklik göstermesidir. Nesnelerin simülasyonunda, gerçek ortamda nesnelere etki eden dış kuvvetler, nesnenin şeklinde değişikliklere yol açar. Bu değişiklikler, dış kuvvetlerin büyüklüklerine göre farklılıklar gösterir. Geçici olarak nesnenin şeklinde bozulmalar meydana gelebilir. Parçalanma, şekil bozukluğu, sıkışma ya da genişleme gibi farklı deformasyonlara maruz kalabilir. Bu nedenle nesne modellemesinde bir takım deformasyon tekniklerine ihtiyaç duyulmaktadır. Deformasyonun gerçek zamanlı etkisinin simüle edilebilmesi için nesnelerin fiziksel yapılarından yola çıkılarak hesaplama yöntemleri kullanılmaktadır.

3.2. Fiziksel yapı

Nesne modellemesi oluşturulurken, fiziksel bir yöntem olarak kütle-yay sistemi etkin bir şekilde kullanılmaktadır. Fiziksel yapıda oluşan değişimlerin sanal ortamdaki görüntüsü Şekil 3.1. ile alınarak ifade edildiğinde, her bir düğüm noktasında bir kütlenin olduğu düşünülür ve bu kütlelerin eşdeğer yaylar ile birbirine bağlı olduğu

varsayılmaktadır. Nesneye dışarıdan gelecek olan herhangi bir kuvvet nesneye etki ettiğinde modelin şeklinde bu kuvvetten dolayı bir değişiklik olacağı için bu üçgen model ile oluşturulan nesnedeki yayların sıkışma, gerilme hareketi yaparak nesnenin deformasyonu sağlanır. Bu şekil değişiklikleri Bölüm 4.1.1'de belirtilen kütle-yay sistemi gerilme, sıkışma potansiyel formülleri aracılığıyla hesaplanmaktadır. Fiziksel yapının bu üçgenlerle birbirine bağlı olduğu düşünüldüğünde, ağ örüntüsünün bu üçgenler ile oluşturulmuş bir biçimde her bir iterasyonda nesnenin şeklinde olan bozulmalara karşı, ayrı bir hesaplama yapılarak deformasyonun kolay bir şekilde oluşturulması sağlanmaktadır.

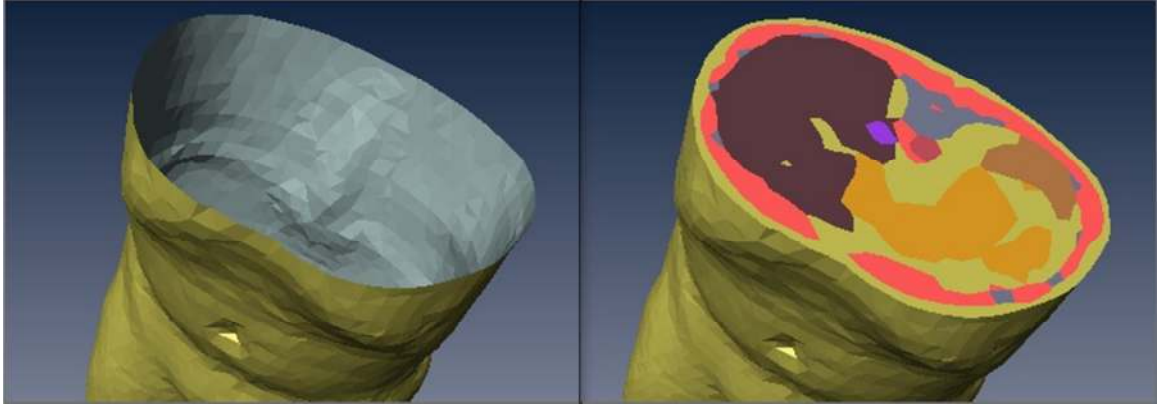


Şekil 3.1. Kütle yay sistemine ait nesne modeli.

1.7.2. Geometrik Şekil

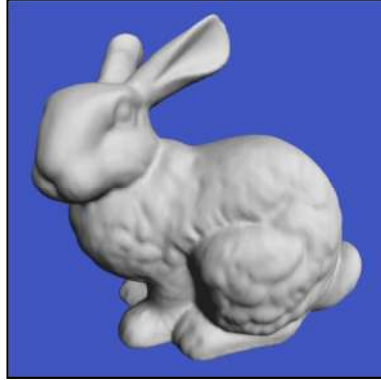
Bir nesnenin simülasyonu gerçekleştirilirken nesne modellemesinde çok farklı geometrik şekillerden yararlanılır. Temel anlamda bir nesnenin gerçek zamanlı simülasyonu gerçekleştirilirken üçgen geometrik şekil ele alınmaktadır. Bunun yanında üçgen geometrik şekilden yola çıkılarak tetrahedral, octahedron ve voxel geometrik şekil geliştirilmiştir. Nesne modellemesinde kullanılan hacimsel modeller nesnenin daha gerçekçi bir görüntüsünün elde edilmesi için kullanılmaktadır. Üçgen geometrik şekilde elde edilen görüntünün içi boş olarak üç boyutlu hacimsel bir görüntüsü elde edilirken, tetrahedral geometrik şekil ile elde edilen modelde nesnenin içi dolu ve gerçekçi bir görüntüsü elde edilir. Bu duruma örnek gösterilebilecek bir insan gövdesi modeli Şekil 3.2(a) ve Şekil 3.2(b) 'de gösterilmiştir.

Bilgisayar grafikleri camiasında simülasyon üzerine çalışmalar yapılarak üçgensel model ile oluşturulan en güzel örnek standford modeli olarak bilinen standford tavşanıdır. Tetrahedral modele en yakın hesapların yapılarak elde edildiği Stanford tavşanı 69451 üçgen yüzeyi ve 34835 ayrı noktadan oluşturularak üçgensel model üzerine kurulmuş örnek bir model olarak karşımıza çıkmaktadır.



Şekil 3.2. a) Triangle geometrik şekil ile oluşturulan içi boş insan gövdesi modeli
b) Tetrahedral geometrik şekil ile oluşturulan içi dolu insan gövdesi modeli

Ayrıca bu çalışmalar sonucunda triangle modele örnek olarak gösterilecek farklı modeller üretilmiş ve üçgen modellemenin başarılı örnekleri sunulmuştur. Bu kapsamda yapılan Ejderha, Asya Dragonu, Mutlu Budha heykeli gibi örnek modeller üretilmiştir. Üçgensel modelleme yöntemi ile oluşturulan Standford Tavşanı ve Mutlu Budha heykeli Şekil 3.3’de ve Şekil 3.4’de gösterilmiştir.



Şekil 3.3. Triangle model ile yapılmış Standford tavşanı.

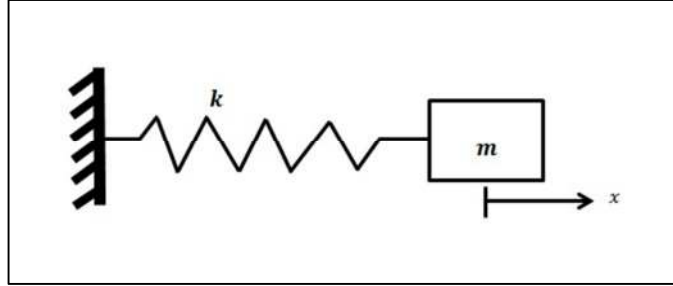


Şekil 3.4. Triangle model ile yapılmış Happy Budha heykeli .

3.1. Hooke Yasası ve Yaylar

Nesnelerin deformasyonu ve esneklik kuramı üzerine ilk ciddi çalışma Robert Hooke (1635-1703) tarafından yapılmıştır. Hooke’a göre bir telin bir ucu bir noktaya bağlanırsa ve diğer ucuna bir ağırlık koyulursa telde bir enerji depolanacak ve bu enerji sayesinde telin boyunda uzama gerçekleşecektir. Ağırlık iki katına çıkarsa, uzama da iki kat artmış olur. Bu durum esneklik sınırına kadar devam eden bir süreçtir. Hooke yasasına göre elastik özelliği olan bir cismin deforme olan miktarının ona etki eden kuvvet ya da gerilim ile doğru orantılı olduğu ifade edilmektedir.

Gerilme-uzama ve esneklik cisimler arasında deęişiklik gösteren özelliklerdir. Uzama ve zorlama doğru orantılı olduęu için bu büyüklüklerin birbirine bölümü deęişmez bir deęer olur. Bu, k esneklik sabiti olarak ifade edilmektedir.



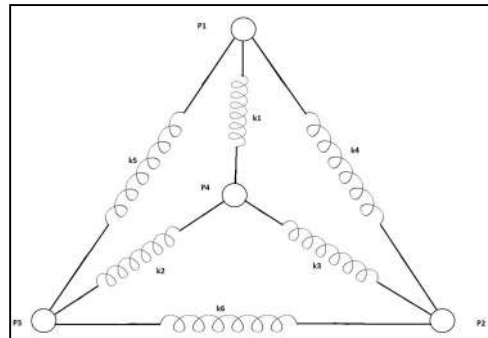
Şekil 3.5. Hooke yasasına göre yay gerilmesi

Hooke yasasına göre Şekil 3.5 ele alındığında, bir yayın ucuna bir kütle bağlanıp bir F kuvveti uygulanması yayın uzamasına yol açar ve etki eden kuvvetin büyüklüğü E.3.1'deki gibi hesaplanır.

$$F = -k * x \quad \text{E.3.1}$$

E.3.1'de geçen F , dışarıdan etki eden kuvveti, k gerilen yaya ait yay sabitini ve x , F kuvvetinin uygulanması sonucu oluşan yer deęiştirmeyi ifade etmektedir.

Sanal ortamda nesnenin simülasyonu gerçekleştirildikten sonra, Kütle-Yay Sistemi (KYS) ile sanal ortamda oluşan kuvvetler, dokunduęu nesneye bir ivme kazandırır ve nesnenin yüzeyinde var olan yaylarda sıkıştırma ya da esneme hareketi oluşturur. Nesne üzerinde uygulanan bu potansiyel kuvvet sayesinde deforme olmuş nesne kendisine etki eden kuvvete zıt yönde bir kuvvet uygular. Nesneye uygulanan kuvvetin nesne yüzeyinde oluşturduęu güç birikimi, nesnenin deformasyon sonucunda var olacak olan yeni şeklini alıncaya kadar devam eder. Kütle-yay sisteminde nesnenin bu potansiyel enerjiyi depolaması ile nesneye etki eden toplam kuvvet hesaplanır. Tetrahedral bir modelde kütle-yay sistemine ait gösterim Şekil 3.6'da gösterilmiştir.



Şekil 3.6. Bir tetrahedral modele ait kütle-yay sistemi.

Kütle-Yay Sisteminde her bir nokta üzerinde oluşan toplam kuvvet, noktaya bağlı olan yay potansiyel kuvveti ile dışarıdan gelen kuvvetlerin toplamına eşittir.

$$f_{toplaml} = f_k + f_d(k) + f_{dış} \quad (E.3.2.)$$

E.3.2'de belirtilen f_k yayların sıkışmasına bağlı olan kuvvetleri, $f_d(k)$ yay sönüm kuvveti, aynı zamanda yaylara ait olan toplam iç kuvveti ve $f_{dış}$ dış çevreden nesneye etki eden dış kuvvetleri göstermektedir. Nesnede dışarıdan bir etki ile oluşan iç kuvvetler, E.3.3'deki gibi hesaplanmaktadır.

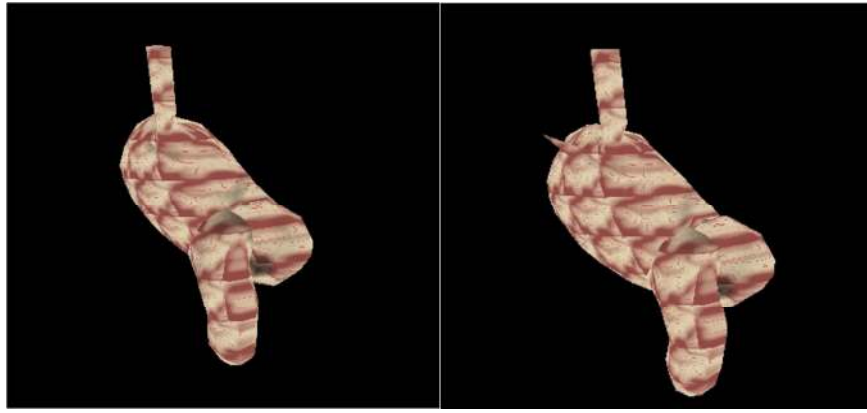
$$f_{int} = f_k + f_d(k) = \sum [k (r - r_0) - d (\|V_a - V_b\|) \frac{x_b - x_a}{\|x_b - x_a\|}] \quad (E.3.3)$$

E.3.3'de belirtilen x_a ve x_b kütle-yayların pozitif kuvvet vektörlerini, v_a ve v_b kütle-yay sistemine ait yaylara ait yaylara ait potansiyel enerjilerin türevlerinin alınmasıyla elde edilen kuvvetlerden hesaplanan hız vektörlerini, r_0 ise kütle ve yaylar arasındaki ilk uzaklıkları, r ise yayların sıkışma ya da uzaması ile elde edilen kütle ve yaylar arasındaki son uzaklığı, k ve d sabitleri ise yay ve sönümleme sabitlerini ifade etmektedir.

Kütle-yay sisteminde nesnenin simülasyonu gerçekleştirildikten sonra dışarıdan gelecek etkilere karşı esnek bir yapıda olması deformasyonun daha net bir biçimde ortaya çıkmasını sağlamaktadır. Nesneye etki eden kuvvetler ile esneklik katsayısı göz önünde bulundurulması, esnekliğin fazla olduğu nesnelerde deformasyonun oluşmasını daha kolay gerçekleştirecektir. Bu durum ayrıca nesnenin deforme edilmesine yönelik kuvvetlerin hesaplanmasında ayrı öneme sahiptir. Kütle-yay sisteminde tetrahedral geometrik şeklin her bir noktasında bir kütlenin var olduğu varsayılarak bu kütleler arasında bir yay sisteminin olduğu ilkesi yer almaktadır.

Nesneye dışarıdan gelecek her bir kuvvet sayesinde oluşacak olan deformasyon değişiklikleri bu nesnelerin yay gerilmeleri ya da sıkışması sonucu depolanan enerjiler ve bu enerjilerin oluşturduğu kuvvetler baz alınarak yeniden sanal ortamda modellenmesini ifade eder. Yaylara bağlı olarak tetrahedral geometrik modelin her bir noktasında bulunan kütlelerin yer değiştirmeleri, yaylar arasında bulunan esneklik katsayısına bağlı olarak değişiklik gösterir. Esneklik değeri küçük ise daha az yer değiştirme olacaktır, ancak esneklik değeri büyükse yer değiştirme daha fazla olacaktır.

Yaylara ait olan gerilme, sıkışma ve sönümlleme hareketleri deformasyonun sanal ortamda gösteriminde kullanılan özelliklerdir. Bir yayın bir F kuvveti ile uzaması yaya bağlı olan kütlelerin yer değiştirmesi anlamına gelmektedir ve bu kapsamda yapılan kuvvet hesaplamaları nesneye ait bir deformasyonu ifade etmektedir. Sanal platformda nesnenin her bir ucunda olduğu düşünülen kütlelerin yer değiştirmesi şekil deformasyonu sağlar ve bu sayede üç boyutlu ortamda nesnenin deformasyon sonrası son halinin gösterilmesi sağlanır. Bu kapsamda yaylara bağlı kütlelerin yer değişikliği kontrol edilerek elde edilen bir mide modelinin deformasyon öncesinde ve sonrasında oluşan görüntüleri Şekil 3.7’de gösterilmiştir.



Şekil 3.7. Tetrahedral şekil ile modellenmiş mide modeli (solda)
Deformasyona uğramış bir mide modeli. (sağda)

3.2. Kütle-Yay Sisteminde Yaylar

Yaylar, kütle-yay sisteminde kütleler arasındaki dengeyi sağlar. Yayların uzaması, sıkışması ya da tamamen eski haline dönmesi yayların bu hareketleri ile mümkün olur. Yayların bağlı atomlar arasındaki yer değişiklikleri ve Hooke yasasına göre yer değiştirmelerinin hesaplanması, yaylara ait özelliklerin kullanılması ile doğru orantılı bir özellik taşımaktadır.

Yaylarda aynı büyüklükte aynı maddeden yapılmış, ancak farklı kalınlıktaki iki yayın yer değiştirme miktarları birbirinden farklı olacaktır. Yayın ne kadar ince olduğu düşünülürse yayın yer değiştirme miktarı o kadar fazla olacaktır. Yayların cinsi, yer değiştirme miktarını etkileyen diğer bir unsurdur. Yaylara ait farklı maddeden yapılmış ancak aynı boyda ve aynı uzunlukta olma özellikleri ne kadar birbirine eşit olsa da, yayların yer değiştirmeleri, farklı maddelerden yapılmış olmalarından dolayı farklı

olacaktır. Ancak Kütle-Yay Sisteminde yaylar eşdeğer olduğu varsayılarak materyal özellikleri k ve d sabitleri ile verilir.

Sanal ortamda nesne simülasyonu gerçekleştirilirken yayların aynı cins ve kalınlıkta olduğu varsayılarak kütle-yay sistemine ilişkin hesaplamalar yapılır. Birbirine bağlı yaylar, her bir noktadan yola çıkılarak birbirine bağlı olduğu varsayılır ve bu noktalar sayesinde nesnenin simülasyonu ve deformasyonu gerçekleştirilir.

3.3. Euler ve Verlet Yöntemi

Euler Yöntemi birinci dereceden bir bilinmeyenli bir yöntemdir. Euler yöntemi birinci dereceden bir bilinmeyenli bir problemin sonucunda yaklaşık bir değer verir ve global hata büyüklüğü ile orantılı bir yöntemdir. Ancak sanal ortamda model deformasyonundaki iterasyonlarda tek başına yeterli olmayan yöntemdir. Temel olarak Euler yöntemine yardımcı olacak olan ikinci dereceden denklem ise Verlet metodudur. E.3.4'de Verlet metoduna ait eşitlik belirtilmektedir. (Şekercioğlu A.S., Duysak A., 2009)

$$X(t + \Delta h) = 2x(t) - x(t - \Delta h) + \Delta h^2 a + O(\Delta h^4) \quad (E.3.4.)$$

$$V(t) = \frac{x(t+\Delta h) - x(t-\Delta h)}{2\Delta h} + O(\Delta h^2)$$

E.3.4'de belirtilen Verlet metodu t anındaki kuvvet ve pozisyonları kullanır. E.3.4'de göre x , t zamanda $t - \Delta h$ ile belirtilen ilk zamana göre, $t + \Delta h$ ile belirtilen son zamanda elde edilen yer değiştirmeyi, V ise bu zaman aralığında kütlenin kazandığı hızı ifade eder. $O(\Delta h)$ parametresi ise kütlelerde oluşan yer değiştirmelere ait hata oranını ifade etmektedir.

Simülasyonda her bir iterasyon sonucunda deformasyona uğrayan nesnenin her bir tetrahedral noktasında bulunan atoma ait hız vektörlerinin hesaplanması gerekmektedir. Buna bağlı olarak, her bir iterasyonda oluşan yer değiştirme vektörü hesaplanmakta ve bunun sonucunda modelde oluşan deformasyona ait koordinat bilgilerinin yeniden hesaplanması sağlanmaktadır.

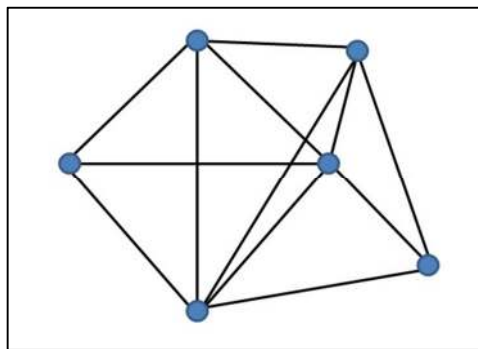
4. MOLEKÜLER MODELLEME

4.1. Yüzey Moleküler Modelleme

4.1.1. Yüzey Üçgen Modelleme

Modelleme fiziksel ve analitik sürece bağlı bir nesnenin sanal ortamda analitik ve sayısal olarak yeniden inşa edilmesidir. Modellemenin gerçekleştirilmesi için modele ait olan problem, güzel bir şekilde analiz edilmelidir. Ayrıca modellemede bu alanla ilgili olarak bilgisayar hafızasına ait olan boş ve gereksiz hesaplamalardan kaçınılması gerekmektedir. Bu nedenle bir modelin deformasyonuna ait sınırlar, iyi bir şekilde belirlenmelidir.

Moleküler modelleme, moleküler yapı , model inşası ve birleşimi için kullanılan fonksiyonlar üzerinde çalışan bir bilimdir. Moleküler modelleme, bir nesnenin sanal ortamda moleküler düzeyde simülasyonu üzerinde çalışır. Bu nesne basit bir anlamda bir plastik ya da metal nesne olabilir. Moleküler modellemede, maddenin moleküler yapısından yola çıkılarak, nesnelerin dışarıdan deformasyona uğraması ile oluşan iç ve dış kuvvetlerin hesaplanmasını da göz önünde bulundurarak, nesnelerin deformasyon sonrası yeniden modellenmesini de ifade etmektedir. Ayrıca fiziksel potansiyel formüllerden yola çıkılarak nesnenin en son hali, deforme olmuş nesnenin iç ve dış kuvvetleri hesaplanarak simüle edilmektedir. Doğada tetrahedral geometrik şekil ile birbirine bağlı bulunan maddelerden esinlenerek elde edilen hacimsel moleküler modellemeye ilişkin atomlar arası bağların gösterildiği temel bir örnek Şekil 4.1’de belirtilmiştir.



Şekil 4.1. Atomlar arası bağlara ilişkin örnek model

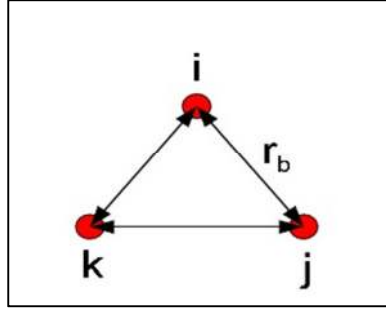
4.1.1.1. Yüzeysel Olarak Modellenmiş Sistemlerde Moleküler Modelleme Metodu

Kütle-yay sisteminde deformasyon öncesinde bir modelde var olan bir potansiyel enerji ile, deformasyon sırasında ve deformasyon sonucu oluşan potansiyel enerjilerin

hesaplanarak modele etki eden toplam kuvvetlerin hesaplanması gerekmektedir. Bu amaçla yapılan fiziksel hesaplamalar nesnenin deformasyon sonrasında şekil değişikliğine uğrayan modelin görüntülenmesi için hesaplanmaktadır.

4.1.1.2. Yüzey-Açı Potansiyel Enerjisi

Üçgensel modellerde bağlı uçlarda temsil edilen atomik etkileşimler ile oluşan yüzey-açı potansiyel enerjisinin gösterimi Şekil 4.2 ‘de verilmiştir.



Şekil 4.2. Deformasyon öncesi denge durumunda olan kütle yay modeli.

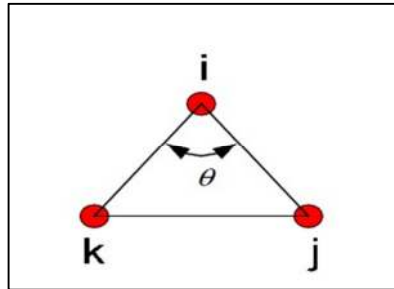
Şekil 4.2’de gösterilen modelde birbirine bağlı yaylar ve atomlar gösterilmiştir. Hooke Kanunu olarak adlandırılan bu modeldeki uçlara ait, yüzeydeki gerilme ile oluşan potansiyel enerjiler E.4.1’deki gibi hesaplanmaktadır.(Şekercioğlu A.S., Duysak A., 2009)

$$\Phi_b = \sum_{1,2 \text{ nokta çifti}} \frac{1}{2} k_b (r_b - r_0)^2 \quad (\text{E.4.1.})$$

E.4.1’de k_b esneklik sabitini, r_b deformasyon sonrası bağ uzaklığını, r_0 deformasyon öncesindeki bağ uzaklığını temsil etmektedir.

4.1.1.3. Bağ-Açı Potansiyel Enerjisi

Bağ açısının sıkışma enerjisi olarak belirtilen bağ-açı potansiyel enerjisi, deformasyon ile üçgenlerin tepe açılarındaki değişim ile elde edildiği potansiyel enerjidir. Şekil 4.3’de üçgen köşelerinde varsayılan 3 atomun oluşturduğu açı olarak tanımlanmaktadır. Bu değişimlerden oluşan potansiyel enerji E.4.2’de belirtilmiştir.



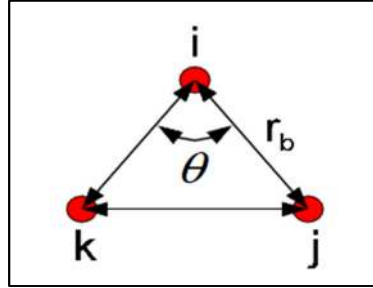
Şekil 4.3. Deformasyon ile birlikte oluşan bağ-açı potansiyel enerjisine ait kütle yay modeli.

E.4.2’de belirtilen k_θ açısı sabitini, $\cos \theta$ deformasyon sonrası oluşan bağ açısı olan θ açısının $\cos$ değerini, $\cos \theta_0$ ise deformasyon öncesi var olan denge bağ açısı olan θ_0 açısının $\cos$ değerini ifade etmektedir. Denge açısı üzerindeki herhangi bir değişiklik bağ-açı potansiyel enerjisinin değerini değiştirecektir. (Duysak A., Şekercioğlu A.S., 2009)

$$\Phi_\theta = \sum_{uc} \frac{1}{2} k_\theta (\cos \theta - \cos \theta_0)^2 \quad (E.4.2)$$

4.1.1.4. Gerilme-Sönümlenme Sonucu Oluşan Potansiyel Enerji

Bir iterasyonluk potansiyel enerji değişimi, iki nokta arasında gerilme ve bükülme açısı olarak formulize edilen bağ açı kuvvetlerindeki değişiklikleri ifade etmektedir. Bir iterasyonluk gerilme ve sönümlenme enerjisini ifade eden denklem E.4.4’de verilmiştir.



Şekil 4.4. Deformasyonda gerilme-sönümlenmeye ait kütle-yay modeli.

E.3.4’de belirtilen a ve b ortak bir atoma bağlı gerilme-sönümlenmeye bağlı oluşan noktaları ifade etmektedir. (Duysak A., Şekercioğlu A.S., 2009)

$$\Phi_{b,\theta} = \sum_{triangle} (\cos \theta - \cos \theta_0) [(r - r^0)_a + (r - r^0)_b] \quad (E.4.3)$$

4.1.1.5. Toplam İç Kuvvet ve Sistem Dinamiği

Bir kütle noktasına ait bir parçacığın toplam potansiyel enerjisi, bağlı atomlar arasındaki potansiyel enerjilerin toplanması ile elde edilir ve elde edilen toplam kuvvet E.4.4’de belirtilmiştir. (Duysak A., Şekercioğlu A.S., 2009)

$$\Phi = \Phi_b + \Phi_\theta + \Phi_{b,\theta} \quad (E.4.4)$$

E.4.4’de belirtilen Φ_b değeri yüzey-açı potansiyel enerjisini, Φ_θ değeri bağ-açı potansiyel enerjisini, $\Phi_{b,\theta}$ değeri ise gerilme-sönümlenme sonucu oluşan potansiyel enerjiyi ifade etmektedir. Kütle yay sistemi ile anlık değişimler sonucunda bir nesneye

etki eden toplam enerji E.4.4' de hesaplanarak, deformasyon sonucunda elde edilen yeni modelin toplam iç kuvvetleri hesaplanır.

Enerjinin türevi kuvveti verir. Bir nesneye bir kuvvet etki ettiğinde oluşan toplam potansiyel enerjinin türevi o nesneye ait olan toplam iç kuvveti verir. Yüzeysel moleküler modellemede nesneye dışarıdan bir kuvvet etki ettiğinde, elde edilen toplam kuvvetin hesaplanması E.4.5'de verilmiştir.

$$F_{toplam} = F_{bağ} + F_{yüzeysel-açı} \quad (E.4.5)$$

Yüzey-açı potansiyel enerjisinin türevi alınarak üçgensel modellemede elde edilen yüzey-açı kuvvetleri, Şekil 4.2 ele alınarak E.4.6'daki gibi hesaplanır.

$$F_{yüzeysel-açı} = -\nabla_b \phi_b = -k_b(r_b - r_0) \vec{r}_b \quad (E.4.6.)$$

$$F_{yüzeysel-açı} = -k_b(r_{ij} - r_{ij}^0) \vec{r}_{ij} + k_b(r_{ik} - r_{ik}^0) \vec{r}_{ik} + k_b(r_{jk} - r_{jk}^0) \vec{r}_{jk}$$

E.4.6'da belirtilen k_b değeri modellenen nesneye ait esneklik sabitini, $r_{ij}^0, r_{ik}^0, r_{jk}^0$ değerleri deformasyon öncesi atomlar arası bağ uzunluğunu, r_{ij}, r_{ik}, r_{jk} deformasyon sonrası atomlar arası yeni bağ uzunluklarını, $\vec{r}_{ij}, \vec{r}_{ik}, \vec{r}_{jk}$ vektörel değerleri ise üç boyutlu ortamda deformasyonun oluşturulduğu vektörel yönü ifade etmektedir.

Bağ-açı potansiyel enerjisinin türevi alınarak üçgensel modellemede elde edilen bağ-açı kuvvetleri, Şekil 4.4 ele alınarak E.4.7'deki gibi ifade edilmektedir.

$$F_{bağ-açı} = -\nabla_\theta \phi_\theta = -k_\theta [\cos \theta - \cos \theta_0] \frac{r_{jk}}{r_{ik} \times r_{ij}} \vec{r}_{jk} \quad (E.4.7)$$

E.4.7'de belirtilen k_θ değeri, üçgen tepe açlarına bağlı olan açı sabitini, r_{jk}, r_{ik}, r_{ij} değerleri üçgen kenar uzunluklarını, $\cos \theta$ deformasyon sonrasında oluşan açının $\cos$ değerini ve $\cos \theta_0$ deformasyon öncesinde denge konumundayken var olan ilk açının $\cos$ değerini, $\vec{r}_{jk}$ değeri ise deformasyona bağlı olarak elde edilen deformasyon yönünü belirten vektörel büyüklüğü ifade etmektedir.

E.4.7'de belirtilen $\cos$ açılarının hesaplanması ise, E.4.8'deki gibi bulunmaktadır.

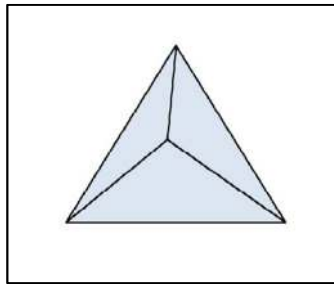
$$r_{jk}^2 = r_{ij}^2 + r_{ik}^2 - 2r_{ij} \times r_{ik} \times \cos \theta$$

$$\cos \theta = \frac{r_{ij}^2 + r_{ik}^2 - r_{jk}^2}{2r_{ij} \times r_{ik}} \quad (E.4.8.)$$

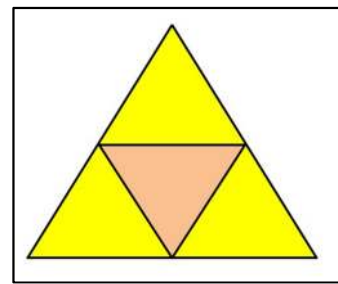
4.2. Hacimsel Moleküler Modelleme

4.2.1. Tetrahedral Modelleme

Geometride düzgün dört yüzlü olarak da ifade edilen tetrahedral model dört üçgen yüzden oluşan bir polihedronudur. Tetrahedral model dört üçgenin eşkenar olduğu bir dörtyüzlüdür. Dörtyüzlü, bir tür piramittir. Piramit, çokgen bir tabanı tek bir noktada birleştiren üçgen yüzlerden oluşur. Dörtyüzlü durumunda taban bir üçgendir (dört yüzün herhangi biri taban sayılabilir), dolayısıyla dörtyüzlü ayrıca üçgen piramit olarak da ifade edilmektedir. Şekil 4.5'de tetrahedral geometrik şekil, Şekil 4.6'da ise tetrahedral geometrik şeklinin üçgen yüzeylerini gösteren tek boyutlu görüntüsü gösterilmiştir.



Şekil 4.5. Tetrahedral geometrik şekil.



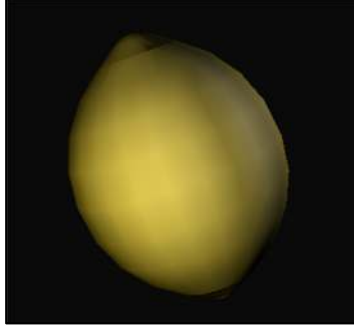
Şekil 4.6. Düzgün dört yüzlü (yüzeyler).

Tetrahedral geometrik modelin her bir noktası ağ oluşumunda birbirine bağlı noktalar olarak modelleme yapılır. Bu modelleme ile tetrahedral geometrik modelin üçgen yüzeyleri birden fazla kullanılabilir. Modelde yüzeysel çakışmalar dikkate alınarak, nesnelerin simülasyonu gerçekleştirilir.

Tetrahedral modelleme, bir nesneye ait simülasyonun tetrahedral geometrik şekil temelinde gerçekleştirilen bir modelleme yöntemidir. Bu kapsamda oluşturulan modelin, hacimsel olarak daha iyi sonuçların elde edildiği gözlemlenmiştir. Tetrahedral modelleme, dört yüzünde var olan üçgen geometrik şekiller ile üçgensel modelleme tekniklerinin entegre edilmesiyle oluşturulan bir tekniktir. Tetrahedral modellemede, simülasyonun ilk aşamasında görüntülenecek olan nesne modelinin tetgen bilgileri ve bu tetgenlerin koordinat düzleminde belirtilen (x,y,z) koordinatlarının bilinmesi gerekmektedir. Bu kapsamda öncelikli olarak modelin çizimi gerçekleştirilir ve bu aşamadan sonra deformasyon işlemlerine geçilir. Şekil 4.7'de bu kapsamda elde edilmiş bir limon modeli gösterilmiştir.

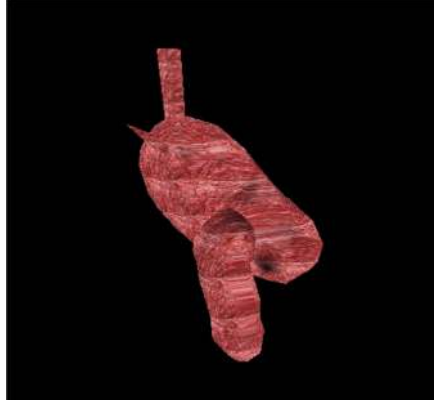
Nesne deformasyonun görüntülenmesi için, moleküler modelleme temelinde kütle-yay sistemi ile Euler ve Verlet metotları kullanılır ve her bir zaman diliminde

simülasyon adımları gözlemlenir. Bu amaçla yapılan fiziksel hesaplamalar simülasyonu oluşturulan modelin deformasyonunda kullanılır.



Şekil 4.7. Tetrahedral modelleme ile oluşturulmuş bir limon modeli

Her bir iterasyon sonucunda deformasyona uğrayan model, gerçek zamanlı olarak gözlemlenir ve gerçekçi bir deformasyonun sağlanmasında iterasyon sonuçları dikkate alınır. Öncelikle tetrahedral bir modelde ilk deformasyonun oluşturulması bir noktanın yer değiştirmesi ilkesinden yola çıkılarak ele alınır, ilk iterasyonda bir noktanın yer değişikliği sanal ortama aktarılır. Bir sonraki iterasyonda ise bu noktaya bağlı olan diğer uçlar kontrol edilerek deformasyon görüntülenir. Bu kapsamda ele alınan ve tetrahedral modelleme yöntemi ile oluşturulmuş olan bir mide modelinin ilk deformasyona uğramış görüntüsü Şekil 4.8’de gösterilmiştir.



Şekil 4.8. Tetrahedral model ile oluşturulmuş bir mide modelinin ilk deformasyona uğramış hali.

Tetrahedral modellemede birbirine bağlı noktalar, kuvvet hesaplamaları sonucunda yer değişikliğine uğrar. Bir noktanın deformasyona uğratılması sonucu diğer noktaların yer değiştirmesi ile oluşan yeni koordinatlar, her bir iterasyon sonucunda tekrardan hesaplanan kuvvet hesaplamaları ile belirlenir. Bu amaçla nesneye etki eden başlangıç kuvveti ile birlikte ilk iterasyonda var olan yer değiştirmeler ile her bir iterasyonda tetrahedral geometrik modele ait kuvvetlerin tamamı yeniden hesaplanmalı

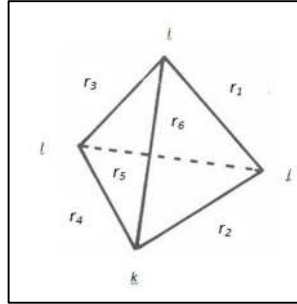
ve bu kuvvet hesaplamaları sonucunda elde edilen yer değiştirmeler sanal ortama aktarılmalıdır.

4.2.1.1. Tetrahedral Modellemede Kütle-Yay Sistemi

Tetrahedral Modellemede yüzey ve bağ açısı kuvvetleri bölüm 3.1.1’de verilen üçgen modellemede kullanılan yöntemlerle bulunur. Bu bağlamda bir deformasyon öncesi, yüzey kuvvetleri hesaplanırken tetrahedral geometrik şekillere ait yüzeylerde bulunan üçgenlerden yararlanılır. Tetrahedral hacimsel modellemede bir deformasyona yönelik iç kuvvetler, yüzey kuvvetleri ve bağ-açı kuvvetlerinin toplamı ile bulunur. Tetrahedral geometrik model ile oluşturularak modellenen bir nesnenin bir noktasına bir kuvvet uygulandığında elde edilen kuvvetler, tetrahedral geometrik şeklin her bir yüzeyinde bulunan üçgen yüzeyler kullanılarak bulunmaktadır.

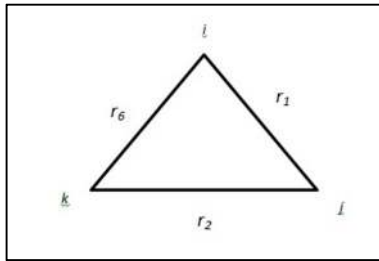
4.2.1.1.1. Tetrahedral Modellemede Yüzey Kuvvetleri

Kütle-yay sisteminde tetrahedral modelde oluşan yüzey kuvvetleri her bir üçgen yüzeyine bağlı olarak değişen yüzeylerde oluşan kuvvetleri kapsamaktadır. Bu amaçla yüzey kuvvetlerine ilişkin gösterim Şekil 4.9 ele alınarak, E.4.9’da gösterilmiştir.



Şekil 4.9. Tetrahedron yüzeylerine ait gösterim

1. Üçgen yüzeye ait kuvvetleri toplamı :

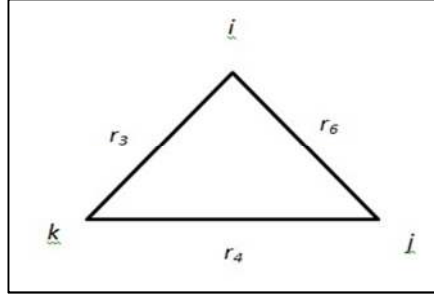


Şekil 4.10. 1. Üçgen yüzeye ait gösterim

$$f_{yüzey1} = -k_b (r_1 - r_1^0) \times \vec{r}_1 - k_b (r_2 - r_2^0) \times \vec{r}_2 - k_b (r_6 - r_6^0) \times \vec{r}_6 \quad (E.4.5)$$

E.4.5'de belirtilen k_b değeri, modellenen nesnenin elastığı özelliğinden elde edilmiş bir esneklik sabitidir. r_1 , r_2 , r_6 uzunlukları ise tetrahedral geometrik modelin 1.üçgen yüzeyine ait üçgen kenar uzunluklarıdır.

2. üçgen yüzeye ait yüzey kuvvetleri toplamı :

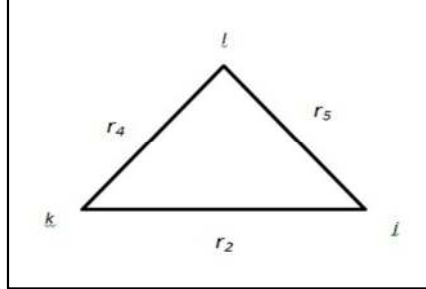


Şekil 4.11. 2. Üçgen yüzeye ait gösterim

$$f_{yüzey2} = -k_b (r_6 - r_6^0) \times \vec{r}_6 - k_b (r_3 - r_3^0) \times \vec{r}_3 - k_b (r_4 - r_4^0) \times \vec{r}_4 \quad (E.4.6)$$

E.4.6'da belirtilen k_b değeri, modellenen nesnenin elastığı özelliğinden elde edilen bir esneklik sabitidir. r_6 , r_3 , r_4 uzunlukları ise tetrahedral geometrik modelin 2.üçgen yüzeyine ait üçgen kenar uzunluklarıdır.

3. üçgen yüzeye ait yüzey kuvvetleri toplamı :



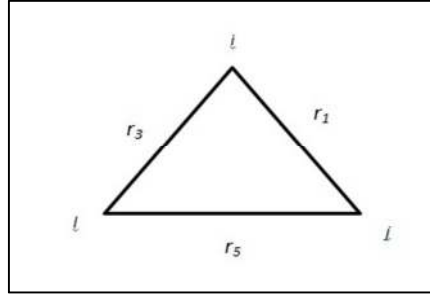
Şekil 4.12. 3. Üçgen yüzeye ait gösterim

$$f_{yüzey3} = -k_b (r_5 - r_5^0) \times \vec{r}_5 - k_b (r_4 - r_4^0) \times \vec{r}_4 - k_b (r_2 - r_2^0) \times \vec{r}_2 \quad (E.4.7)$$

E.4.7'de belirtilen k_b değeri, modellenen nesnenin elastığı özelliğinden elde edilen bir esneklik sabitidir. r_5 , r_4 , r_2 uzunlukları ise tetrahedral geometrik modelin 3.üçgen yüzeyine ait üçgen kenar uzunluklarıdır.

4. üçgen yüzeye ait yüzey kuvvetleri toplamı :

$$f_{yüzey4} = -k_b (r_1 - r_1^0) \times \vec{r}_1 - k_b (r_3 - r_3^0) \times \vec{r}_3 - k_b (r_5 - r_5^0) \times \vec{r}_5 \quad (E.4.8.)$$



Şekil 4.13. 4. Üçgen yüzeye ait gösterim

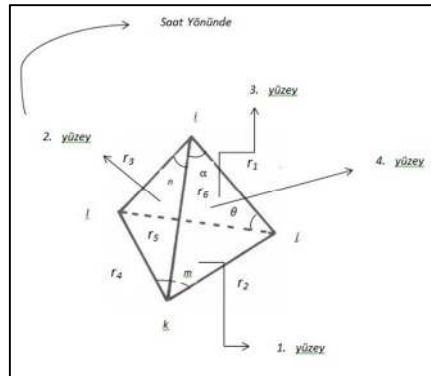
E.4.8'da belirtilen k_b değeri, modellenen nesnenin elastığı özelliğinden elde edilen bir esneklik sabitidir. r_1, r_3, r_5 uzunlukları ise tetrahedral geometrik modelin 4.üçgen yüzeyine ait üçgen kenar uzunluklarıdır.

$$F_{\text{yüzeyToplamı}} = -k_b (r_1 - r_1^0) \times \vec{r}_1 - k_b (r_2 - r_2^0) \times \vec{r}_2 - k_b (r_6 - r_6^0) \times \vec{r}_6 - k_b (r_3 - r_3^0) \times \vec{r}_3 - k_b (r_5 - r_5^0) \times \vec{r}_5 - k_b (r_4 - r_4^0) \times \vec{r}_4 \quad (\text{E.4.9.})$$

E.4.9'da belirtilen k_b değerleri tetrahedral modellemede ile modellenen nesneye ait olan modelin elastik özelliğine bağlı olan esneklik sabitidir. r değerleri deformasyon sonrası üçgen yüzeylerde oluşan yeni üçgen kenar uzunluk değerleri ve r_0 olarak belirtilen değerler ise başlangıçtaki üçgen kenar uzunluk değerleridir. $\vec{r}$ vektörel değerleri ise deformasyonun üç boyutlu ortamda oluşan şekil değişikliğinin yönünü belirlemek amacıyla kullanılmaktadır. Her bir yüzeyde tekrar eden kuvvetler, yeniden hesaplanmaması için toplam yüzey kuvvetine dahil edilmez. Bu sayede nesneye etki eden kuvvetler daha gerçekçi ve doğru bir hesaplama ile hesaplanmaktadır.

4.2.1.2. Tetrahedral Modellemede Bağ-açı Kuvvetleri

Tetrahedral modellemede kütle-yay sistemi ile oluşan bağ-açı kuvvetleri tetrahedral geometrik şekle ait üçgen yüzeylerin açıları ve kenar uzunluklarından yararlanılarak elde edilen kuvvetlerin toplamıdır. Bu kapsamda Şekil 4.14 ele alınarak elde edilen bağ açı kuvvetleri E.4.8'de gösterilmiştir.



Şekil 4.14. Tetrahedral modellemede üçgen yüzeylere ait deformasyon sonucu oluşan şekle bağlı bağ açı kuvvetleri değerleri

E.4.11’de belirtilen k değerleri yüzey kuvvetlerinin hesaplanmasında da kullanılan modellenen nesneye ait esneklik sabiti, kosinüs değerleri ise Şekil 4.14’de belirtilen üçgen yüzeylerine ait açılara ait $\cos \theta$ değerleri, deformasyon sonucu oluşan $\cos$ değerleri ve r değerleri de deformasyon öncesi üçgen yüzey kenarlarına ait kenar uzunluklarını ifade etmektedir. $\vec{r}$ vektörel değerleri ise deformasyonun hangi yönde değişeceğini belirten r değerlerine ait koordinat bilgileridir.

$$\cos m = \frac{r_4^2 + r_2^2 + r_5^2}{2 \times r_2 \times r_4}$$

$$\cos n = \frac{r_3^2 + r_6^2 + r_4^2}{2 \times r_3 \times r_6}$$

$$\cos \theta = \frac{r_1^2 + r_5^2 + r_3^2}{2 \times r_1 \times r_5}$$

$$\cos \alpha = \frac{r_1^2 + r_6^2 + r_2^2}{2 \times r_1 \times r_6}$$

E.4.10

$$\begin{aligned} F_{\text{bağcıtoplamı}} &= k \times (\cos m - \cos m_0) \times \frac{-r_5}{r_2 \times r_4} \times \vec{r}_5 + \\ &k \times (\cos n - \cos n_0) \times \frac{-r_4}{r_3 \times r_6} \times \vec{r}_4 + \\ &k \times (\cos \theta - \cos \theta_0) \times \frac{-r_3}{r_1 \times r_2} \times \vec{r}_3 + \\ &k \times (\cos \alpha - \cos \alpha_0) \times \frac{-r_5}{r_2 \times r_4} \times \vec{r}_5 \end{aligned} \quad (\text{E.4.11})$$

E.4.10’de belirtilen $\cos$ değerleri, bir deformasyon sonrası tetrahedral geometrik şeklin üçgen yüzeylerine ait tepe açılarından elde edilen $\cos$ değerlerdir. E.4.11’de belirtilen $\cos \theta$ değerleri ise deformasyondan önce üçgen yüzeylerin ilk açılara ait ve ilk uzunluklara bağlı olarak bulunan $\cos \theta$ değerleridir.

4.3. Hacimsel Kuvvet Hesabı

4.3.1 Biss-Wass Haman Potansiyel Enerji Fonksiyonu

Biss-Wass Haman Potansiyel Enerji Fonksiyonu (PEF) kütle, yüzey, katmanlı ve kendi dokuları arasındaki yapılar için uyarlanmıştır. Kütlenin ve silikon kümesinin yapısal özelliklerini inceler. Biss-Wass Haman Potansiyel Enerji fonksiyonu E.4.12' deki gibidir. (Erkoç Ş., 1996)

$$\Phi = \phi_2 + \phi_3 = \sum_{i<j} U_{ij} + \sum_{i<j<k} W_{ijk}, \quad (\text{E.4.12})$$

$$U_{ij} = f_c(r_{ij})[A_1 e^{\lambda_1 r_{ij}} + A_2 e^{\lambda_2 r_{ij}}], \quad W_{ijk} = V_3(r_{ij}, r_{ik}, \theta) f_c(r_{ij}) f_c(r_{ik}),$$

$$V_3(r_{ij}, r_{ik}, \theta) = B_1 \psi_1(r_{ij}) \psi_1(r_{ik}) \left(\cos \theta + \frac{1}{3} \right)^2 + B_2 \psi_1(r_{ij}) \psi_1(r_{ik}) \left(\cos \theta + \frac{1}{3} \right)^3,$$

$$f_c(r) = [1 + e^{(r-r_c)/\mu}]^{-1}, \quad \psi_i(r) = e^{-\alpha_i r^2},$$

E.4.12'de belirtilen $A_1, A_2, B_1, B_2, \lambda_1, \lambda_2$ ve μ değerleri deformasyona uğrayan nesnenin kimyasal yapısına ait parametrelerdir. r değerleri nesnenin tetrahedral yapısına bağlı olarak tetrahedral geometrik şeklin yüzey kenarlarını ifade etmektedir. θ açısı ise tetrahedral geometrik şeklin üçgen yüzeylerindeki tepe açısı olarak ifade edilmektedir.

Bu potansiyel enerji fonksiyonu tetrahedral yapı özelliği için uygun bir potansiyel enerji fonksiyonudur. Bu kapsamda elde edilen potansiyel enerjilere ait türevler modellenen nesneye ait kuvvet hesaplamalarında kullanılmak üzere elde edilen eşitlikleri kapsar.

4.3.1.1 Biss-Wass Haman Potansiyel Enerji Fonksiyonu ile Elde Edilen Kuvvetler

Biss-Wass potansiyel enerji fonksiyonu, maddelerin tetrahedral geometrik molekül yapısından elde edilmiş olan potansiyel enerji fonksiyonlarıdır. Yapılan tez çalışmasında, biss-wass potansiyel enerji fonksiyonu kullanılarak nesne modellemesinde bir nesneye etki eden iç kuvvetlerin hesaplanması için, biss-wass potansiyel enerji fonksiyonunun türevi alınarak, kuvvet fonksiyonu elde edilmiştir. Bu kapsamda Biss-Wass Haman potansiyel enerji fonksiyonunun türevinin elde edilmesi ile nesne modellemesinde kullanılmak üzere bu çalışmada elde edilen kuvvetler E.4.13. ve E.4.14'deki gibidir.

$$\Phi' = \phi_2' + \phi_3' = \sum_{i<j} U_{ij}' + \sum_{i<j<k} W_{ijk}' \quad (\text{E.4.13})$$

Biss-wass Haman potansiyel enerji fonksiyonu ile elde edilen kuvvetlerin bulunmasında, nesnenin kimyasal yapısını içeren parametrelerin değeri 1 kabul edilerek kuvvetlerin net bir değer hesabının yapılması sağlanmış ve deformasyonun net bir görüntüsü elde edilmiştir.

$$\begin{aligned}
& \sum_{i < j} \left[\frac{- \left(\frac{(r_{ij}-r_0)}{\mu} \right) \times \frac{1}{\mu} \times \vec{r}_{ij}}{\left(\frac{(r_{ij}-r_0)}{\mu} \right)^2} \times (A_1 \times e^{-\lambda_1 \times r_{ij}^2} + A_2 \times \right. \\
& \left. e^{-\lambda_2 \times r_{ij}^2}) + \left[A_1 \times e^{-\lambda_1 \times r_{ij}^2} \times (-2 \times r_{ij} \times \lambda_1) + A_2 \times \right. \right. \\
& \left. \left. e^{-\lambda_2 \times r_{ij}^2} \times (-2 \times r_{ij} \times \lambda_2) \right] \times \frac{1}{\frac{(r_{ij}-r_0)}{\mu}} \right] + \sum_{i < j < k} \left[\left(\beta_1 \times \right. \right. \\
& \left. \left. e^{-\alpha_1 \times r_{ij}^2} \times e^{-\alpha_1 \times r_{ik}^2} \times \left(\cos \theta + \frac{1}{3} \right)^2 + \beta_2 \times e^{-\alpha_2 \times r_{ij}^2} \times \right. \right. \\
& \left. \left. e^{-\alpha_2 \times r_{ik}^2} \times \left(\cos \theta + \frac{1}{3} \right)^3 \right] \times \frac{1}{\frac{(r_{ij}-r_0)}{\mu}} \right] \times \frac{1}{1+e} \\
& \left. - \left(\frac{(r_{ik}-r_0)}{\mu} \right) \times \frac{1}{\mu} \times \vec{r}_{ik} \right) + \left(\left[\beta_1 \times e^{-\alpha_1 \times r_{ij}^2} \times e^{-\alpha_1 \times r_{ik}^2} \times \right. \right. \\
& \left. \left. \left(\cos \theta + \frac{1}{3} \right)^2 + \beta_2 \times e^{-\alpha_2 \times r_{ij}^2} \times e^{-\alpha_2 \times r_{ik}^2} \times \left(\cos \theta + \frac{1}{3} \right)^3 \right] \times \right.
\end{aligned}$$

$$\begin{aligned}
& \left. \frac{1}{1+e^{\frac{(r_{ik}-r_0)}{\mu}}} \times \frac{-\left(e^{\frac{(r_{ij}-r_0)}{\mu}}\right)^{\frac{1}{\mu} \times \overrightarrow{r_{ij}}}}{\left[1+e^{\frac{(r_{ij}-r_0)}{\mu}}\right]^2} + \left(\frac{1}{1+e^{\frac{(r_{ij}-r_0)}{\mu}}} \times \right. \right. \\
& \left. \frac{1}{1+e^{\frac{(r_{ik}-r_0)}{\mu}}} \times \left(\beta_1 \times e^{-\alpha_1 \times r_{ij}^2} \times e^{-\alpha_1 \times r_{ik}^2} \times 2 \times \left(\cos \theta + \right. \right. \right. \\
& \left. \left. \frac{1}{3} \right) \times (-\sin \theta) + \beta_1 \times e^{-\alpha_1 \times r_{ij}^2} \times \left(\cos \theta + \frac{1}{3} \right)^2 \times e^{-\alpha_1 \times r_{ik}^2} \times \right. \\
& \left. (-2 \times \alpha_1 \times r_{ik}) + \beta_1 \times \left(\cos \theta + \frac{1}{3} \right)^2 \times e^{-\alpha_1 \times r_{ik}^2} \times \right. \\
& \left. e^{-\alpha_1 \times r_{ij}^2} \times (-2 \times \alpha_1 \times r_{ij}) + \beta_2 \times e^{-\alpha_2 \times r_{ij}^2} \times e^{-\alpha_2 \times r_{ik}^2} \times \right. \\
& \left. 3 \times \left(\cos \theta + \frac{1}{3} \right) \times (-\sin \theta) + \beta_2 \times e^{-\alpha_2 \times r_{ij}^2} \times \left(\cos \theta + \frac{1}{3} \right)^3 \times \right. \\
& \left. e^{-\alpha_2 \times r_{ik}^2} \times (-2 \times \alpha_2 \times r_{ik}) + \beta_2 \times e^{-\alpha_2 \times r_{ik}^2} \times \left(\cos \theta + \right. \right. \\
& \left. \left. \frac{1}{3} \right)^3 \times e^{-\alpha_2 \times r_{ij}^2} \times (-2 \times \alpha_2 \times r_{ij}) \right) \left. \right) \quad (E.4.14)
\end{aligned}$$

4.3.2. Khor-Dass Potansiyel Fonksiyonu

Khor-Dass Potansiyel Enerji Fonksiyonu (PEF) potansiyel enerji fonksiyonu temel tetrahedral yapılı yarı iletkenleri için geliştirilmiş bir potansiyel enerji fonksiyonudur. Khor-Dass Potansiyel Enerji fonksiyonu E.4.15 'de verilmiştir.

(Erkoç Ş., 1996)

$$\begin{aligned}\Phi &= \phi_{2,3} = \sum_{i<j;k} U_{ij;k}, \\ U_{ij;k} &= A f_{ij} [e^{-\theta} r_{ij} - g_{ij}(1 + h_{ij})], \\ f_{ij} &= e^{-\beta(r_{ij}-R_i)}, \\ g_{ij} &= \frac{B_0}{Z_i^\alpha} e^{-i, r_{ij}}, \\ h_{ij} &= \sum_{k \neq i,j} [\cos \eta (\theta_{ijk} - \theta_i) - 1].\end{aligned}\tag{E.4.15}$$

E.4.15'de belirtilen β , $\frac{B_0}{Z_i^\alpha}$, η değerleri deformasyona uğrayan nesnenin madde yapısına ilişkin parametreler olup, khor-dass enerji fonksiyonun türevi alınırken deformasyonun daha net bir görüntüsün sağlanması için yapılan tez çalışmasında bazı sadeleştirilmeler yapılarak ve bu parametrelerin değerleri 1 kabul edilerek, E.4.16'da belirtilen enerji fonksiyonu elde edilmiştir.

$$\begin{aligned}\Phi &= \phi_{2,3} = \sum_{i<j;k} U_{ij;k}, & U_{ij;k} &= f_{ij} [e^{-\theta} r_{ij} - g_{ij}(1 + h_{ij})], \\ f_{ij} &= e^{-1(r_{ij}-R_i)}, & g_{ij} &= e^{-i, r_{ij}}, & h_{ij} &= \sum_{k \neq i,j} [\cos \eta (\theta_{ijk} - \theta_i) - 1].\end{aligned}\tag{E.4.16}$$

Formülde belirlenmiş parametreler, elmas yapının bağ-düğüm açısının sabitleri ve kütle modüllerinin potansiyel enerji fonksiyonuna uyarlanmasıyla belirlenmiş parametrelerdir. Potansiyel enerji fonksiyonu çeşitli yapılarda bağ enerjisini ve atomlar arası denge uzaklığını verir. Khor-dass potansiyel enerji fonksiyonunda belirtilen eşitlikler potansiyel enerjilerin türevleri alınarak, deforme edilen modele ait kuvvet hesaplamalarında gerekli olan eşitlikleri kapsamaktadır.

Bu kapsamda Khor-dass potansiyel fonksiyonuna ait Karbon, Silisyum ve Germanyum elementleri için formülde verilen parametrelerin karşılıkları Çizelge 3.1' de verilmiştir.

Çizelge 3.1. Karbon, Silisyum ve Germanyum için Khor Dass potansiyel fonksiyonu parametreleri.

Parametre	C	Si	Ge
A (eV)	10237.546	2794.2386	1278.555
B_0	0.938766	0.08251716	0.3481879
θ (Å ⁻¹)	3.20244	3.13269	2.37239
λ (Å ⁻¹)	3.05075	1.34146	1.54775
α	0.08596252	0.6249096	0.381135
β	44.85709	25.44123	17.79861
γ	3.10786	3.38218	3.22877
η	0.1824903	0.90084597	0.6460521

4.3.2.1 Khor-Dass Potansiyel Fonksiyonu İle Elde Edilen Kuvvetler

Khor-Dass Potansiyel Fonksiyonu ile bu çalışma kapsamında enerji formülüne ait eşitliklerin türevlerinin elde edilmesi ile nesne deformasyonuna ilişkin elde edilen kuvvet fonksiyonları E.3.17'deki gibidir.

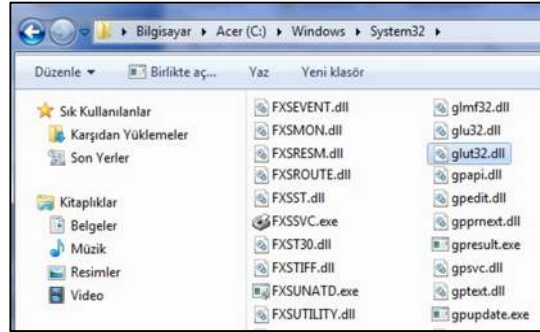
$$\begin{aligned}
 U'_{i,j,k} = & A \times \left(-\beta \times e^{-\beta \times (r_{ij}-r_0)} \times \vec{r}_{ij} \right) \times \left(e^{-\theta \times r_{ij}} - e^{-\lambda \times r_{ij}} \left(1 + \sum_{k \neq i,j} [\cos \eta (\theta_{ijk} - \theta_i) - 1] \right) \right) \\
 & + A \times e^{-\beta \times (r_{ij}-r_0)} \times \left(e^{-\theta \times r_{ij}} \times (-\theta) - [e^{-\lambda \times r_{ij}} \times (-\lambda \times r_{ij}) \times \right. \\
 & \left. (1 + \sum_{k \neq i,j} [\cos \eta (\theta_{ijk} - \theta_i) - 1]) + e^{-\lambda \times r_{ij}} \times (-\sum_{k \neq i,j} [\sin \eta \times (\theta_{ijk} - \theta_i)]) \right)
 \end{aligned}
 \tag{E.3.17}$$

Khor-Dass Potansiyel enerji fonksiyonu ile elde edilen kuvvetlerde, nesne modelinin kimyasal yapısından dolayı nesnenin iç yapısına bağlı olarak elde edilen parametreler, simülasyon ortamında daha net bir görüntünün elde edilmesi için 1 kabul edilmiştir. E.3.17'de belirtilen A , β , η , λ değerleri uygulama aşamasında değerleri 1 olarak kabul edilen parametrelerdir.

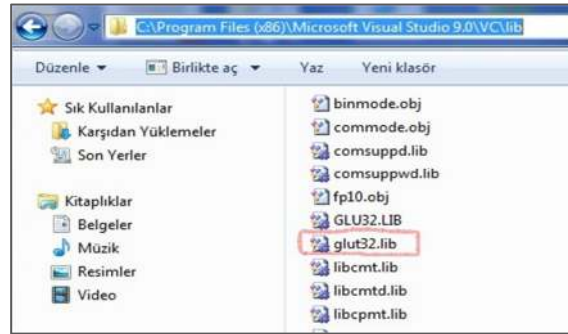
5. DEFORMASYON SİMÜLASYONU

5.1. OpenGL

Bilgisayar grafikleri teknolojileri üzerinde yapılan araştırmalar sonucunda, bir nesnenin sanal ortamda görüntülenebilmesi için bir çok grafik programı geliştirilmiştir. Simülasyon için nesne modellemesinde hesaplamaların yapılabilirdiği C++ programlama diline entegre edilebilen bir kütüphane olan OpenGL kütüphanesi, geliştirilen grafik programlama dillerinden birisidir. Simülasyonun gerçekleşmesi için gerekli olan kütüphane, bu tarz programlama dilleri ile etkin bir şekilde kullanıldığında verimli sonuçların alındığı görülmüştür.

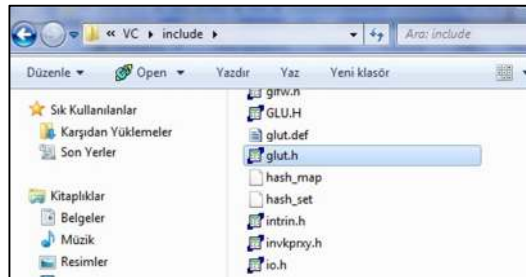


Şekil 5.1. OpenGL kütüphanesine ait glut32.dll dosyasının eklenmesi.

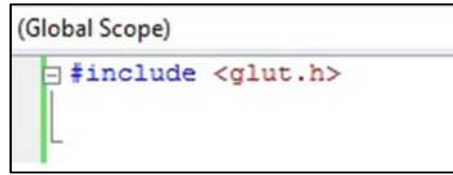


Şekil 5.2. OpenGL kütüphanesine ait glut32.lib dosyasının eklenmesi.

OpenGL nesne görüntüleme dili, C++ programlama diline “glut.h” kütüphanesinin eklenmesi ile görüntülenen bir dildir.



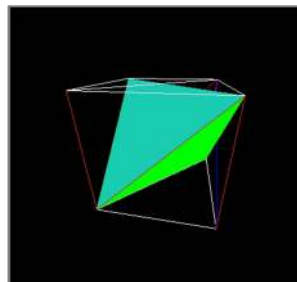
Şekil 5.3. OpenGL kütüphanesine ait glut.h dosyasının eklenmesi.



Şekil 5.4. OpenGL kütüphanesinin C++ platformuna aktarılması.

OpenGL kütüphanesinin C++ programlama editörüne aktarılması için, Şekil 5.1'deki gibi glut32.dll dosyasının "Windows" klasörü içinde bulunan "System32" klasörüne, "glut32.lib" dosyasının Şekil 5.2'deki gibi C++ derleyicisi içinde bulunan "lib" klasörüne ve son olarak "glut.h" dosyasının Şekil 5.3'deki gibi C++ derleyicisine bağlı "include" klasörüne entegre edilmesi gerekmektedir. Ayrıca program komut satırının başında Şekil 5.4.'deki gibi OpenGL kütüphanesi C++ platformuna aktarılmalıdır.

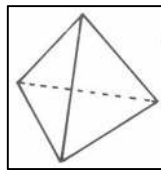
Sistematik olarak üçgen yapılarla elde edilen yüzey modellemesi sayesinde tetrahedral modelin dış yüzeyleri elde edilmekte, OpenGL görüntüleme metotları kullanılarak üçgen yüzeylerin 3 boyutlu görüntüsü elde edilebilmektedir. Bu kapsamda bu çalışmada OpenGL dili ile oluşturulan örnek bir tetrahedral model Şekil-5.5'de gösterilmiştir.



Şekil -5.5. OpenGL dili ile yapılmış bir tetrahedral model.

5.2. Geometrik Yapı

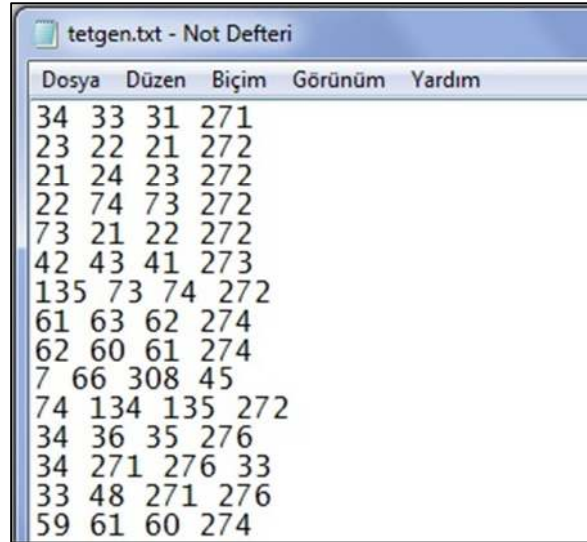
Nesnelerin hacimsel modellenmesinde, üçgen geometrik şekillerden yola çıkılarak, tetrahedral model geliştirilmiştir. Bu model sayesinde gerçek zamanda bir nesnenin 3 boyutlu görüntüsü tetrahedral geometrik şekil ile oluşturulmuş, nesneye dış ortamdan etki eden dış kuvvetlerin oluşturduğu deformasyona ait fiziksel hesaplamalar, bu geometrik şekil üzerinden hesaplanarak deformasyon hesaplamaları yapılmıştır. Bir tetrahedral modele ait olan temel gösterim Şekil 5.6'da gösterilmiştir.



Şekil 5.6. Tetrahedral Geometrik Şekil.

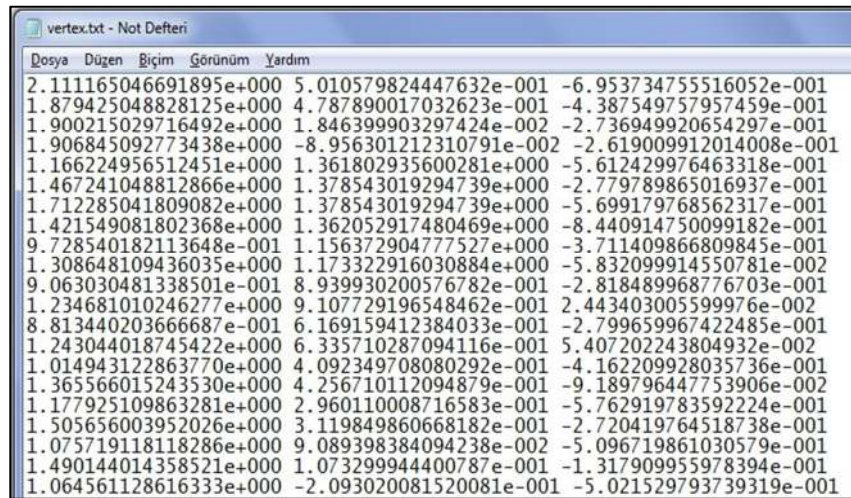
5.3. Modelleme

Tetrahedral geometrik yapı ile OpenGL kütüphanesi ve C++ programlama dili ile oluşturulan simülasyonda tetrahedral modelin her bir ucunda bir kütleinin olduğu varsayılır. Tetrahedral modele ait olan uçlar, numaralandırılmış olarak dosya okuma yöntemi ile bir text dosyasından okutulmaktadır.



Dosya	Düzen	Biçim	Görünüm	Yardım
34	33	31	271	
23	22	21	272	
21	24	23	272	
22	74	73	272	
73	21	22	272	
42	43	41	273	
135	73	74	272	
61	63	62	274	
62	60	61	274	
7	66	308	45	
74	134	135	272	
34	36	35	276	
34	271	276	33	
33	48	271	276	
59	61	60	274	

Şekil 5.7. Tetrahedral modelin nokta bilgilerinin tutulduğu tetgen dosyası.



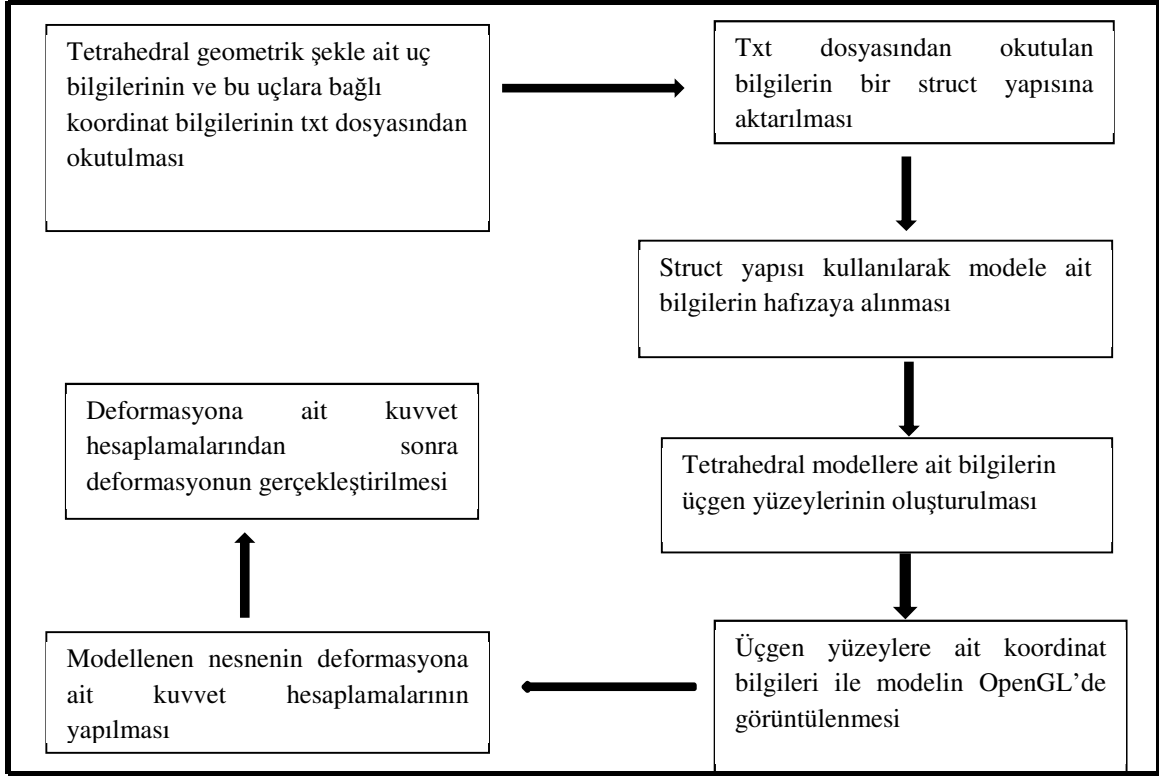
Dosya	Düzen	Biçim	Görünüm	Yardım
2.111165046691895e+000	5.010579824447632e-001	-6.953734755516052e-001		
1.879425048828125e+000	4.787890017032623e-001	-4.387549757957459e-001		
1.900215029716492e+000	1.846399903297424e-002	-2.736949920654297e-001		
1.906845092773438e+000	-8.956301212310791e-002	-2.619009912014008e-001		
1.166224956512451e+000	1.361802935600281e+000	-5.612429976463318e-001		
1.467241048812866e+000	1.378543019294739e+000	-2.779789865016937e-001		
1.712285041809082e+000	1.378543019294739e+000	-5.699179768562317e-001		
1.421549081802368e+000	1.362052917480469e+000	-8.440914750099182e-001		
9.728540182113648e-001	1.156372904777527e+000	-3.711409866809845e-001		
1.308648109436035e+000	1.173322916030884e+000	-5.832099914550781e-002		
9.063030481338501e-001	8.939930200576782e-001	-2.818489968776703e-001		
1.234681010246277e+000	9.107729196548462e-001	2.443403005599976e-002		
8.813440203666687e-001	6.169159412384033e-001	-2.799659967422485e-001		
1.243044018745422e+000	6.335710287094116e-001	5.407202243804932e-002		
1.014943122863770e+000	4.092349708080292e-001	-4.162209928035736e-001		
1.365566015243530e+000	4.256710112094879e-001	-9.189796447753906e-002		
1.177925109863281e+000	2.960110008716583e-001	-5.762919783592224e-001		
1.505656003952026e+000	3.119849860668182e-001	-2.720419764518738e-001		
1.075719118118286e+000	9.089398384094238e-002	-5.096719861030579e-001		
1.490144014358521e+000	1.073299944400787e-001	-1.317909955978394e-001		
1.064561128616333e+000	-2.093020081520081e-001	-5.021529793739319e-001		

Şekil 5.8. Tetrahedral modele ait noktaların koordinatların tutulduğu dosya.

Tetrahedral modele ait olan noktaların her bir noktasına ait koordinatlar ise ayrı bir text dosyasında tutulmaktadır. Modellemenin başlangıcında bu noktalar C++ programlama dilinde okutularak, her bir noktanın koordinatları ile birlikte modelleme gerçekleştirilmektedir. Dosyalarda tutulan her bir tetrahedral geometrik şeklin noktaları ve koordinatları Şekil 5.7 ve Şekil 5.8’de gösterilmiştir.

5.4. Uygulama Adımları

Tetrahedral modellemede nesnenin sanal ortamda gösterilmesi için belli bir düzen içinde ve sistemli bir şekilde görüntülenmesi gerekmektedir. Uygulama adımlarına ilişkin tez çalışmasına yönelik uygulanan adımlar Şekil 5.9'daki gibi sıralayabiliriz.



Şekil 5.9. Tetrahedral Modellemede bir modele ait uygulama adımları

5.5. Nesnelerin sanal ortama aktarılması

Modellemede bir nesnenin simülasyonunun gerçekleştirilmesi için, tetrahedral geometrik şekillerden oluşan noktalar kümesi, C++ programlama dilinde bir struct yapısına aktarılır.

```

// Koordinate bilgisi
typedef struct {
    float x;
    float y;
    float z;
}VertexInfo;

// Tetgen Üçgen yüzey normal bilgisi
typedef struct {
    float nx;
    float ny;
    float nz;
}NormalInfo;

// Tetgen Üçgen yüzeyler için bilgileri
typedef struct {
    int nokta[3];
}TriangleInfo;

// Tetrahedral Uçlara Ait Bilgi
typedef struct {
    int n[4];
}NodeInfo;

// Tetrahedral Uçlara ait uzunluk bilgisi
typedef struct {
    float r[4];
}Kemeruzunluk;

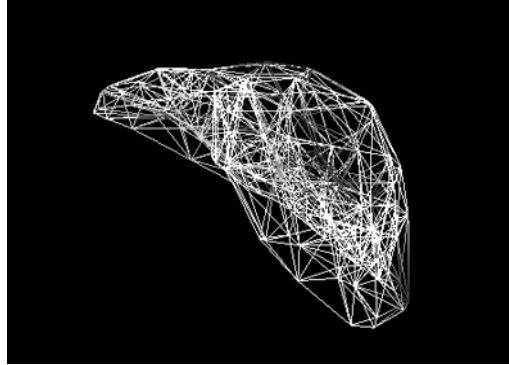
typedef struct {
    float koe[4];
}Koe;
  
```

Şekil 5.10. Tetrahedral geometrik şekillerin tutulduğu struct yapısı.

Bu sayede sistem tasarımı oluşturulurken, nesneye ait olan tetrahedral şekillerin nokta ve koordinat bilgisinin hafızada tutularak, hem nesnenin modellenmesinde hem de nesnenin dışarıdan gelecek dış kuvvetler nedeniyle oluşan deformasyona ait fiziksel hesaplamalarda kullanılması amaçlanmıştır. Bu amaçla oluşturulmuş olan struct yapısı Şekil 5.10'da gösterilmiştir.

5.6. Ağ Oluşumu (Meshing)

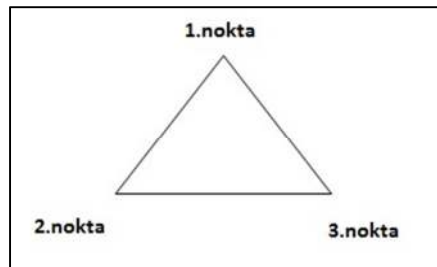
Tetrahedral geometrik şekiller oluşturularak yapılan simülasyon adımlarında ağ oluşumu, nesnenin düzgün ve gerçeğe yakın bir görüntüsünün elde edilmesi açısından önemlidir. Tetrahedral uçların okutulduğu dosyada tetrahedral modelin yüzeyinde bulunan üçgen yüzeylere ait köşe noktalarının tekrarlanmaması gerekmektedir. Tetrahedral ağ oluşumunda yüzeylerin üçgen geometrik şekle sahip olması ağ örüntüsünün oluşumunda üçgen metotların kullanılması modellemeyi kolaylaştıran ayrı bir unsurdur. Bir tetrahedral modellemeye ait olan örnek bir ağ oluşumu Şekil 5.11'de verilmiştir.



Şekil 5.11. Tetrahedral ağ oluşumunun temel bir görüntüsü.

5.7. Normallerin Hesaplanması

Ağ oluşumunda ağ modelinin net bir görüntüsünün elde edilmesi için, modelin alt yapısında kullanılan tetrahedral geometrik modelin uç noktalarının görüntülenmemesi için tetrahedral geometrik şeklin her bir üçgen yüzeyine ait normal vektörlerinin hesaplanması gerekir. Bu sayede modelin dış yüzeyinin düzgünleştirilmesi sağlanır.



Şekil 5.12. Bir üçgenin noktaları.

Tetrahedral modelin yüzeylerine ait üçgenlerin normal vektörlerinin hesaplanmasında kullanılan yöntem, Newell Metodu (Hill F.S., 1999) olarak bilinmektedir. Newell Metodu Şekil 5.12'deki üçgen ele alınarak E.5.1.'de belirtilmiştir.

1. *nokta koordinatları* $(x1, y1, z1)$

2. *nokta koordinatları* $(x2, y2, z2)$

3. *nokta koordinatları* $(x3, y3, z3)$

$$xa = (x1 - x2)$$

$$ya = (y1 - y2)$$

$$za = (z1 - z2)$$

$$xb = (x2 - x3)$$

$$yb = (y2 - y3)$$

$$zb = (z2 - z3)$$

$$xn = (ya \times zb) - (za \times yb)$$

$$yn = (za \times xb) - (xa \times zb)$$

$$zn = (xa \times yb) - (ya \times xb)$$

$$uzunluk = \sqrt{xn^2 + yn^2 + zn^2}$$

$$xno = \frac{xn}{uzunluk}$$

$$yno = \frac{yn}{uzunluk}$$

$$zno = \frac{zn}{uzunluk} \quad (E.5.1.)$$

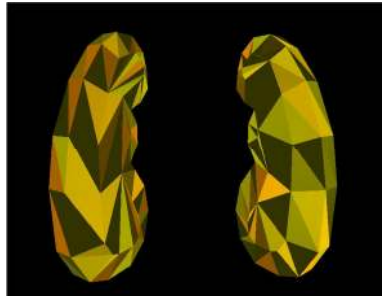
Bu işlemler sonucunda bulunan değerler, OpenGL dilinde normal vektörlerinin sanal ortamda nesne yüzeyinin net bir görüntüsünün oluşmasını sağlayan `glNormal(xno,yno,zno)` değerleri olacaktır.

Normal vektörleri nesnenin sanal ortamda gerçeğe yakın görüntüsünün elde edilmesi için gerekli olan vektörlerdir. Nesnenin gerçekçi bir görüntüsünün elde edilmesi için tetrahedral yüzeye ait birbirine komşu olan üçgen yüzeylerin ve noktaların normallerinin ortalamasının alınması gerekmektedir. Sanal ortamda normal ortalaması alınmadan nesne görüntülendiğinde yüzeyinin pürüzlü ve üçgen uçların görüntülediği bir nesne modeli oluşur. Bu durumda ortalama hesabı yapılmadan oluşan model, gerçeğe yakın bir görüntünün elde edilmesini engellemektedir. Sanal

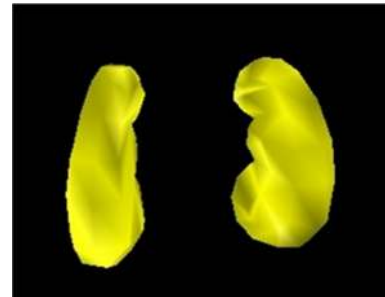
ortamda nesnenin normal vektörlerinin ortalaması hesaplanmadan oluşturulan şekli ile normal ortalaması alınarak yapılan şekli Şekil 5.13’de ve Şekil 5.14’de gösterilmiştir.

Ağ oluşumunda tetrahedral yüzeylere ait olan üçgenler belirlenmelidir. Bu üçgenler, tetrahedral modelde birbirine bağlı olan komşu noktaların birleşiminden oluşmaktadır.

Nesnenin gerçeğe yakın modellenmesi gerçekleştirilirken birbirinin aynısı olan üçgen yüzeylerin modelde tekrar görüntülenmemesi gerekmektedir. Bu yüzeylere ait olan noktalarda, üçgen noktaları olarak ele alınır ve noktalara ait bir sıralama yapılır.



Şekil 5.13. Normal ortalaması alınmadan modellenmiş böbrek modeli



Şekil 5.14. Normal ortalaması alınarak modellenmiş böbrek modeli

Bu sıralama sonucunda birbirinin aynısı olan yüzeyler bir kere okunur ve buna göre normal hesaplamaları yapılır. Tetrahedral modelde komşu üçgen yüzeyler ile bağlantılı noktaların normallerinin alınması ile nesnenin gerçeğe yakın görüntüsünün elde edilmesi sağlanmaktadır. Tetrahedral model ile simüle edilen bir nesnenin tetrahedral yüzeyleri ve bağlantılı uçları ile normallerinin hesaplamaları yapılan ve bu yüzey ve uçlara ait normallerin ortalamaları alınarak gerçeğe yakın görüntüsünün elde edildiği model Şekil 5.14’de gösterilmiştir.

```
//Yüzey Kaplama ve Tetrahedral Yüzeylerin Çizilmesi
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);

for(i=0;i<Tetgenfileline*4;i++)
{
    if (kontrol[i]==0 && noktafrekans[i]>1)
    {
        glBegin(GL_TRIANGLES);
        glNormal3f(VertexNormal[Ucgen[i].nokta[0]-1].nx,VertexNormal[Ucgen[i].nokta[0]-1].ny,VertexNormal[Ucgen[i].nokta[0]-1].nz);
        glVertex3f(Vkoordinat[Ucgen[i].nokta[0]-1].x , Vkoordinat[Ucgen[i].nokta[0]-1].y, Vkoordinat[Ucgen[i].nokta[0]-1].z);
        glNormal3f(VertexNormal[Ucgen[i].nokta[1]-1].nx,VertexNormal[Ucgen[i].nokta[1]-1].ny,VertexNormal[Ucgen[i].nokta[1]-1].nz);
        glVertex3f(Vkoordinat[Ucgen[i].nokta[1]-1].x , Vkoordinat[Ucgen[i].nokta[1]-1].y, Vkoordinat[Ucgen[i].nokta[1]-1].z);
        glNormal3f(VertexNormal[Ucgen[i].nokta[2]-1].nx,VertexNormal[Ucgen[i].nokta[2]-1].ny,VertexNormal[Ucgen[i].nokta[2]-1].nz);
        glVertex3f(Vkoordinat[Ucgen[i].nokta[2]-1].x , Vkoordinat[Ucgen[i].nokta[2]-1].y, Vkoordinat[Ucgen[i].nokta[2]-1].z);
        glEnd();
    }
}

glDisable(GL_TEXTURE);
glutSwapBuffers();
```

Şekil 5.15. Yüzey kaplaması ve tetrahedral modele ait üçgen çizimine ait kod bloğu.

Tetrahedral model ile oluşturulmuş olan nesnelerin düzgün bir görüntüsünün elde edilmesinden sonra, modele uygun bir resim ile yüzeyin kaplanması gerekmektedir. OpenGL programlamada yüzeyin kaplanması ile ilgili olarak bu tezde kullanılan yöntem Şekil 5.15’de gösterilmiştir. Modellenmiş olan nesnelerin uygun bir resimle kaplanması gerçekçi bir görüntünün elde edilmesini sağlar. Bu amaçla kullanılan texture mapping yöntemi ile nesneye ait bir görüntünün elde edilmesi sağlanmaktadır. OpenGL’de doku resminin bütün modeli ve tetrahedral geometrik şeklin yüzeylerini kolayca kaplaması açısından bmp uzantılı olan ve boyutu küçük olan bir doku resmi seçilir. Boyut olarak küçük bir resim seçilmesi modellemenin hızı açısından kullanıcılara büyük bir avantaj sağlar.

5.8. Modellenen Nesnede Deformasyon

Sanal ortamda tetrahedral geometrik şekil ile hacimsel modellenmiş nesnenin dışarıdan oluşabilecek etkilere karşı şekil değişikliğine uğraması öngörülmektedir. Buna bağlı olarak nesneye dışarıdan bir F kuvveti uygulandığında bu kuvvetin nesneye etki ettiği noktada değişiklik olması beklenir.

Hacimsel moleküler modelleme ve kütle-yay sistemi ile dışarıdan etkilenen modele ait olan toplam iç kuvvetler hesaplanır. Toplam iç kuvvetin hesaplanması ile ilk deformasyona ait yer değiştirmeler hesaplanır. Modellenmiş nesneye dışarıdan bir kuvvet uygulandığında oluşacak olan yer değiştirme algoritması Şekil 5.16’da gösterilmiştir. Bir noktaya bir F kuvveti uygulandığında ilk iterasyonda nesneye ait bir noktanın yer değiştirmesi gözlenir. Buna bağlı olarak her bir iterasyon için toplam iç kuvvet yeniden hesaplanır.

```
float deformasyon(float deg_X,float deg_Y,float deg_Z,int degisenNokta)
{
    for(i=0;i<Tetgenfileline*4;i++)
    {
        secim=1;
        for(p=0;p<3;p++)
        {
            if (Ucgen[i].nokta[p]==degisenNokta)
            {
                Vkoordinat[Ucgen[i].nokta[p]].x=Vkoordinat[Ucgen[0].nokta[1]].x+(deg_X);
                Vkoordinat[Ucgen[i].nokta[p]].y=Vkoordinat[Ucgen[0].nokta[1]].y+(deg_Y);
                Vkoordinat[Ucgen[i].nokta[p]].z=Vkoordinat[Ucgen[0].nokta[1]].z-(deg_Z);
            }
        }
    }
    return 1;
}
```

Şekil 5.16. Deformasyon sonucu bağlantılı noktaya etki eden yer değiştirmelere ait kod bloğu.

Tetrahedral geometrik şekle bağlı komşu diğer tetrahedral geometrik şekiller bir ya da birden fazla nokta ile birbirine komşu olabilir. Komşu uçların, nesnenin bir

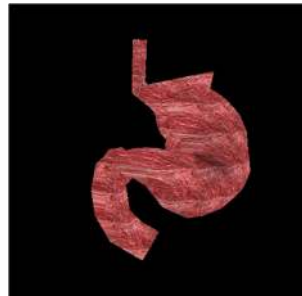
noktasına etki eden F kuvveti ile yer değiştirmelerin bulunması için birbirine komşu olan ve tekrar etmeyen noktaların kontrolü gerekmektedir. Bu tez çalışmasında bu işlemi gerçekleştiren algoritma bloğu Şekil 5.17’de gösterilmiştir.

```
void baglantiliTetgenler(int baglantiliNokta)
{
    int i=0;
    Uclar=( NodeInfo *)malloc(Tetgenfileline*sizeof( NodeInfo));
    // Tetgenleri tekrar okutuyorum...
    TetgenRead(Tetgenfile,uc1,uc2,uc3,uc4);
    for(i=0;i<Tetgenfileline;i++)
    {
        Uclar[i].n[0]=uc1[i];
        Uclar[i].n[1]=uc2[i];
        Uclar[i].n[2]=uc3[i];
        Uclar[i].n[3]=uc4[i];
        tetgenNo[i]=1;
    }
    for(i=0;i<Tetgenfileline;i++) //Tetgen sayisi kadar
    {
        if( Uclar[i].n[0]==baglantiliNokta || Uclar[i].n[1]==baglantiliNokta ||
        Uclar[i].n[2]==baglantiliNokta || Uclar[i].n[3]==baglantiliNokta )
        {
            if ( Uclar[i].n[0]==baglantiliNokta )
            {
                Tetgendiger3nokta[tetgenbaglantilिनokta].nd[0]=Uclar[i].n[1];
                Tetgendiger3nokta[tetgenbaglantilिनokta].nd[1]=Uclar[i].n[2];
                Tetgendiger3nokta[tetgenbaglantilिनokta].nd[2]=Uclar[i].n[3];
                tetgenNo[i]=i;
            }
            if ( Uclar[i].n[1]==baglantiliNokta )
            {
                Tetgendiger3nokta[tetgenbaglantilिनokta].nd[0]=Uclar[i].n[0];
                Tetgendiger3nokta[tetgenbaglantilिनokta].nd[1]=Uclar[i].n[2];
                Tetgendiger3nokta[tetgenbaglantilिनokta].nd[2]=Uclar[i].n[3];
                tetgenNo[i]=i;
            }
            if ( Uclar[i].n[2]==baglantiliNokta )
            {
                Tetgendiger3nokta[tetgenbaglantilिनokta].nd[0]=Uclar[i].n[0];
                Tetgendiger3nokta[tetgenbaglantilिनokta].nd[1]=Uclar[i].n[1];
                Tetgendiger3nokta[tetgenbaglantilिनokta].nd[2]=Uclar[i].n[3];
                tetgenNo[i]=i;
            }
            if ( Uclar[i].n[3]==baglantiliNokta )
            {
                Tetgendiger3nokta[tetgenbaglantilिनokta].nd[0]=Uclar[i].n[0];
                Tetgendiger3nokta[tetgenbaglantilिनokta].nd[1]=Uclar[i].n[1];
                Tetgendiger3nokta[tetgenbaglantilिनokta].nd[2]=Uclar[i].n[2];
                tetgenNo[i]=i;
            }
            tetgenbaglantilिनokta=tetgenbaglantilिनokta+1;
        }
    }
    flag[baglantiliNokta]=1;
    TetgenDigerUclarıKontrolEt(baglantiliNokta);
    tetgenbaglantilिनokta=0;
}
```

Şekil 5.17. Bir tetrahedral geometrik şekle bağlı olan uçları kontrol eden kod bloğu

5.9. Khor-Dass ve Biss-Wass Haman Potansiyel Fonksiyonu İle Elde Edilen Kuvvetlere Euler ve Verlet Metodlarının Uygulanması

Khor-Dass ve Biss-Wass Haman Enerji fonksiyonlarını kullanarak elde edilen kuvvetler, Bölüm 2.4’de belirtilen Euler ve Verlet yöntemleri kullanılarak, ivme ve hızın bulunması ve buna bağlı olarak modelde oluşan yer değiştirmenin hesaplanması ile deformasyonun sağlandığı görülmüştür. Bu kapsamda elde edilen bir modelinin deformasyona uğramış bir görüntüsü Şekil 5.18’de gösterilmiştir.



Şekil 5.18. Biss-wass haman ve Khor-dass kuvvetleri ile oluşturulmuş bir deformasyon örneği

Khor-dass potansiyel enerji fonksiyonu ve biss-wass haman enerji fonksiyonu ile elde edilen kuvvetlerin simülasyona uygulanması aşamasında nesne modelinin kimyasal yapısından elde edilen parametreler, 1 kabul edilerek yeni bir yaklaşım oluşturulmuş ve bu sayede hesaplamaların daha hızlı ve görüntünün daha net elde edilmesi sağlanmıştır. Bu kapsamda simülasyon uygulamasında kullanılan khor-dass ve biss-wass haman enerji fonksiyonlarından elde edilen kuvvetlerin bu çalışmada oluşturulmuş olan algoritmaları Şekil 5.19(a), Şekil 5.19(b) ve Şekil 5.20'de gösterilmiştir.

```
float biss_wass_haman_kuvveti()
{
    aci *bhucg;
    sinus *bhsin;
    kenaruzunluk *bhuz;
    bhucg=(aci *)malloc(Tetgenfileline*4*sizeof(aci));
    bhsin=(sinus *)malloc(Tetgenfileline*4*sizeof(sinus));
    bhuz=(kenaruzunluk *)malloc(Tetgenfileline*12*sizeof(kenaruzunluk));
    ucgen_uzunluklari(bhuz);
    sinus_hesabi(bhsin,bhuz);
    if (secim==1)
    {
        ucgen_uzunluklari(defbhuz);
    }

    for(int i=0;i<Tetgenfileline;i++)
    {
        Uij=Uij+
            (((-1*(eussu(-1,(defbhuz[i].r[0]-bhuz[i].r[0]/mu))))
            /
            (pow( (1+ (eussu(-1,(defbhuz[i].r[0]-bhuz[i].r[0]/mu))), 2 ))) *
            (( balfa1 * ( eussu(lamda1,pow(bhuz[i].r[0],2)))
            +
            ( balfa2 * ( eussu(lamda2,pow(bhuz[i].r[0],2))))
            +
            (((balfa1*( eussu(lamda1,pow(bhuz[i].r[0],2)))*(-
            2*bhuz[i].r[0]*lamda1) )+(balfa2*( eussu(lamda2,pow(bhuz[i].r[0],2)))*(-
            2*bhuz[i].r[0]*lamda2) )) *
            (1 / (1+ (eussu(-1,(defbhuz[i].r[0]-bhuz[i].r[0]/mu)))))) ;
        Vuc=Vuc+
            ((betal*eussu( alfa1, (pow(bhuz[i].r[0],2))) * eussu( alfa1,
            (pow(bhuz[i].r[5],2))) * 2 * ( tekkosinus(bhucg,bhuz,i) + (1/3) ) * (-1)*
            (teksinus(bhsin,bhuz,i) ) )
            +
            (betal* eussu( alfa1, (pow(bhuz[i].r[0],2))) * pow(
            tekkosinus(bhucg,bhuz,i) + (1/3) ,2) * eussu( alfa1, (pow(bhuz[i].r[5],2)))
            * (-2) * alfa1 * bhuz[i].r[5] )
            +
            (betal* pow( (tekkosinus(bhucg,bhuz,i) + (1/3)),2) * eussu( alfa1,
            (pow(bhuz[i].r[5],2))) * eussu( alfa1, (pow(bhuz[i].r[0],2))) * (-2) * alfa1
            * bhuz[i].r[0] )
            +
            (beta2*eussu( alfa2, (pow(bhuz[i].r[0],2))) * eussu( alfa2,
            (pow(bhuz[i].r[5],2))) * 3 * ( tekkosinus(bhucg,bhuz,i) + (1/3) ) * (-1)*
            (teksinus(bhsin,bhuz,i) ) )
            +
            (beta2* eussu( alfa2, (pow(bhuz[i].r[0],2))) * pow(
            tekkosinus(bhucg,bhuz,i) + (1/3) ,3) * eussu( alfa2, (pow(bhuz[i].r[5],2)))
            * (-2) * alfa2 * bhuz[i].r[5] )
            +
            (beta2* eussu( alfa2, (pow(bhuz[i].r[5],2))) * pow(
            tekkosinus(bhucg,bhuz,i) + (1/3) ,3) * eussu( alfa2, (pow(bhuz[i].r[0],2)))
            * (-2) * alfa2 * bhuz[i].r[0] ));
    }
```

Şekil 5.19.(a) Biss-wass Haman kuvvetlerinin hesaplanmasına yönelik programın 1. blok parçası.


```

Wijk=Wijk+
{
    (beta1* (eussu(alfa1, pow(bhuz[i].r[0],2))) * (eussu(alfa1,
    pow(bhuz[i].r[5],2))) * pow ( ((tekkosinus(bhucg,bhuz,i) + (1/3))),2 ) )
    +
    (beta2* (eussu(alfa2, pow(bhuz[i].r[0],2))) * (eussu(alfa2,
    pow(bhuz[i].r[5],2))) * pow ( ((tekkosinus(bhucg,bhuz,i) + (1/3))),3 ) )
    )
    *
    ( ( 1 / (1+ eussu(-1, defbhuz[i].r[0]-bhuz[i].r[0]/mu) ) *
    -1*
    ( eussu(-1, defbhuz[i].r[5]-bhuz[i].r[5]/mu) * 1/mu ) / pow(( 1 +
    eussu(-1, defbhuz[i].r[5]-bhuz[i].r[5]/mu)),2 ) )
    +
    ((beta1* (eussu(alfa1, pow(bhuz[i].r[0],2))) * (eussu(alfa1,
    pow(bhuz[i].r[5],2))) * pow ( ((tekkosinus(bhucg,bhuz,i) + (1/3))),2 ) )
    +
    (beta2* (eussu(alfa2, pow(bhuz[i].r[0],2))) * (eussu(alfa2,
    pow(bhuz[i].r[5],2))) * pow ( ((tekkosinus(bhucg,bhuz,i) + (1/3))),3 ) ) )
    *
    ((1 / ( 1 + eussu(-1, defbhuz[i].r[0]-bhuz[i].r[0]/mu)) )
    *
    -1*
    ( eussu(-1, defbhuz[i].r[0]-bhuz[i].r[0]/mu) * 1/mu ) / pow(( 1 +
    eussu(-1, defbhuz[i].r[0]-bhuz[i].r[0]/mu)),2 ) )
    +
    ( ( 1 / ( 1 + eussu(-1, defbhuz[i].r[0]-bhuz[i].r[0]/mu)) ) * ( 1
    / ( 1 + eussu(-1, defbhuz[i].r[5]-bhuz[i].r[5]/mu) ) * Vuc )));
    phi=(phi)+ (Uij + Wijk);
}
printf("\n Phi degeri-> %f\n",phi);
return phi;
}

```

Şekil 5.19.(b) Biss-wass Haman kuvvetlerinin hesaplanmasına yönelik programın 2. blok parçası.

```

float khor_dass_kuvveti()
{
    aci *pfucg;
    sinus *pfsin;
    kenaruzunluk *pfuz;
    pfucg=(aci *)malloc(Tetgenfileline*4*sizeof(aci));
    pfsin=(sinus *)malloc(Tetgenfileline*4*sizeof(sinus));
    pfuz=(kenaruzunluk *)malloc(Tetgenfileline*12*sizeof(kenaruzunluk));
    ucgen_uzunluklari(pfuz);
    sinus_hesabi(pfsin,pfuz);
    for(int i=0;i<Tetgenfileline;i++)
    {
        Uijk = Uijk + (
        (balfa*(-1*beta*eussu(beta,pfuz[i].r[0]))
        *
        (eussu(teta,pfuz[i].r[0])-
        (eussu(lamda,pfuz[i].r[0])*(1+kosinus_hesabi(pfucg,pfuz))))
        +
        (balfa*eussu(beta,pfuz[i].r[0]))
        *
        (
        (eussu(teta,pfuz[i].r[0])*(-1)*teta)-
        (
        eussu(lamda,pfuz[i].r[0])*(1+kosinus_hesabi(pfucg,pfuz)+(eussu(lamda,pfuz[i].r[0])*(-1*(
        sinus_hesabi(pfsin,pfuz)))))))));
    }
    return (Uijk);
}

```

Şekil 5.20. Khor-dass kuvvetlerinin hesaplanmasına yönelik program bloğu.

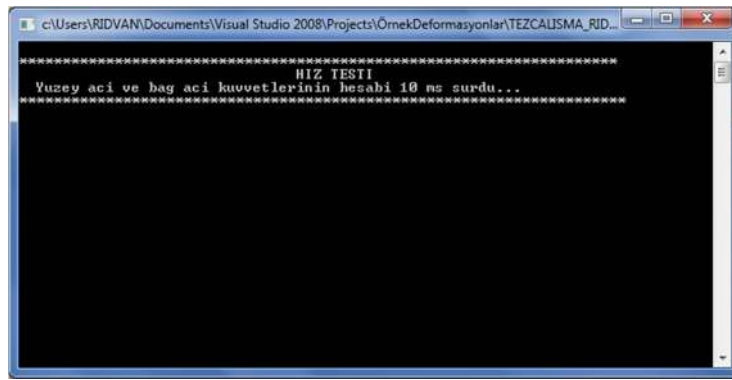
5.10. Moleküler Modelleme ve Hacimsel Modelleme ile Yapılan Deformasyonun Hız Testleri

Yapılan tez çalışmasında, hacimsel moleküler modelleme ile elde edilen bağ-açı ve yüzey kuvvetleri ile elde edilen deformasyon görüntüleri ile biss-wass haman ve khor-dass kuvvetlerinden elde edilen görüntülerin hız testleri oluşturulmuştur. Bu kapsamda biss-wass haman ve khor-dass potansiyel enerji fonksiyonları ile elde edilen kuvvet hesaplamalarında, bağ-açı ve yüzey-açı kuvvetlerinin hesaplanması ile mukayese edildiğinde, daha fazla hesaplama yapılmasından dolayı, zamanlama olarak

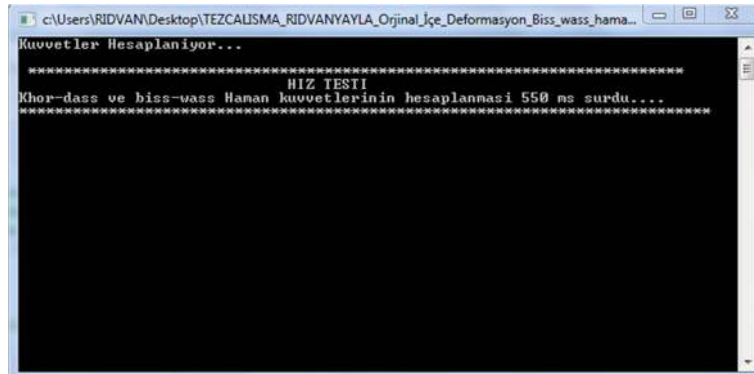
hızının daha yavaş olduğu ancak deformasyonun daha net görüntülediği gözlemlenmiştir.



Şekil 5.21. Hız testinde kullanılan karaciğer modelinin yüzey-açı ve bağ-açı kuvvetleri (solda) ve biss-wass haman ve khor-dass kuvvetleri (sağda) ile elde edilen deformasyon görüntüsü



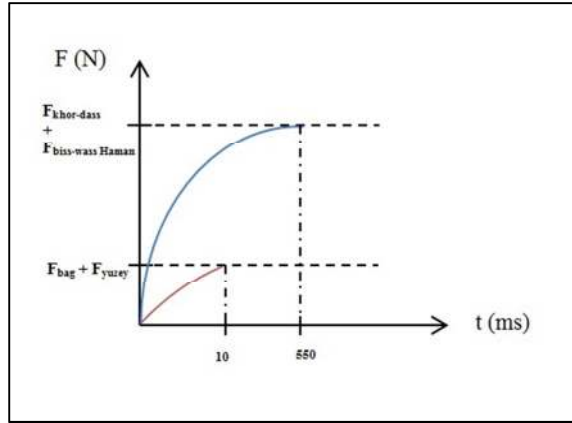
Şekil 5.22. Moleküler modelleme ile yapılan deformasyonda geçen süre



Şekil 5.23. Khor-dass ve Biss-wass Haman kuvvetleri ile yapılan deformasyonda geçen süre

Bu kapsamda, deformasyona uğratılmış bir karaciğer modeli ele alınmış ve iki ayrı program üzerinde yapılan hız testleri sonucunda bağ-açı ve yüzey açı kuvvetleri hesaplanarak, Bölüm 3.3’de belirtilen euler ve verlet yöntemlerinin kullanımı ile elde edilen deformasyonun 10 ms, khor-dass ve biss-wass haman kuvvetlerinin hesaplanarak euler ve verlet yöntemlerinin kullanımı ile elde edilen deformasyonun 550 ms’lik bir zamanla elde edildiği saptanmıştır. Bu bağlamda zamanlama olarak biss-wass ve khor-dass kuvvetlerinin hesaplanmasında daha fazla sürenin geçtiği ancak daha net bir

deformasyonun sağlandığı görülmüştür. Bu kapsamda elde edilen veriler ile oluşan deformasyonlar, Şekil 5.21’de belirtilen aynı karaciğer modeli üzerinde gösterilmiş olup, test sonuçları Şekil 5.22 ve Şekil 5.23’de gösterilmiş ve grafiksel gösterimi ise Şekil 5.24’de belirtilmiştir.



Şekil 5.24. Hız testinde moleküler modelleme ve hacimsel modellemede geçen süreyi gösteren grafik

6. SONUÇ VE ÖNERİLER

Modellemenin günümüz teknolojisinde hemen hemen her alanda kullanılması, modelleme tekniklerinde yeni yaklaşımların, yeni algoritmaların gelişimine katkıda bulunmuştur. Bu alanda yapılan yenilikçi yaklaşımlar, simülasyonlar için nesnelerin fiziksel ve kimyasal boyutta incelenmesini sağlamış, disiplinler arası bir çalışmayı gerektiren incelemelerin sanal ortamda uygulanmasının önünü açmıştır.

Yapılan tez çalışmasında nesnelerin tetrahedral geometrik şekil temelinde üç boyutlu hacimsel görüntüleri elde edilmiştir. Bu kapsamda yapılmış olan çalışmalardan yola çıkılarak, elde edilen simülasyonlarda kütle-yay sistemi ve moleküler modelleme ışığında nesnelerin deformasyon simülasyonu gerçekleştirilmiş, tetrahedral geometrik şeklinin üçgen yüzeyleri kullanılarak, üçgensel modelleme yöntemleri tetrahedral modellemede kullanılmıştır. Ayrıca nesnelerin hacimsel boyutta simülasyonu gerçekleştirilmiş ve hacimsel moleküler modelleme yöntemleri ile deformasyon simülasyonları gözlemlenmiştir. Ayrıca nesnelerin kimyasal bağ yapılarından ve moleküler yapısından yola çıkılarak elde edilen potansiyel enerji fonksiyonları deformasyonda kullanılmış ve bu fonksiyonların türevleri elde edilerek, deformasyon simülasyonu için gerekli olan kuvvet fonksiyonları elde edilmiştir.

Sanal bir ortamda iki boyutlu gerçekleştirilen simülasyonların, ihtiyaçların analizinde yeterli olmadığı alanlarda, üç boyutlu modelleme tekniklerinin uygulanması farklı alanlardaki ihtiyaçlara karşılık verilmesine olanak sağlamıştır. Moleküler modellemede nesnelerin, atomik yapısının incelenerek, atomlar arası bağların ve bu bağlardan kaynaklanan etkileşimlerin simülasyon tekniklerine uyarlanarak yeniden formülize edilmesi, üç boyutlu hacimsel modellemenin sanal ortamda daha fazla anlaşılır olmasına katkıda bulunmuştur. Maddelerin moleküler geometrik şekillerinden yola çıkılarak oluşturulmuş olan simülasyon modelleri üçgensel modelleme temelinde, hacimsel moleküler modellemede daha sağlıklı bir görüntünün elde edilmesini sağlamıştır.

Tetrahedral geometrik şekil, üçgen yüzeyleri sayesinde kütle-yay sistemi ve moleküler modelleme ile oluşturulan yöntemlerin hacimsel moleküler modellemede kullanılmasını sağlamıştır. Tetrahedral geometrik şekil ile oluşturulan simülasyonlar , fiziksel boyutta ve kimyasal boyutta incelendiğinde sanal ortamda oluşturulan

deformasyonların daha nitelikli ve kaliteli bir şekilde oluşturulmasını sağladığı görülmüştür. Hacimsel modelleme teknikleri ile tetrahedral geometrik şekil, üç boyutta nesnelerin daha gerçekçi incelenmesini sağlamaktadır. Bu kapsamda modellenen cisimlerde oluşan deformasyon değişiklikleri, kullanıcılara daha gerçekçi bir ortam sağlamış ve daha dinamik bir görüntünün elde edildiğini göstermiştir. Euler metodu ile diferansiyel denklemlerin kullanıldığı bir çok yöntemde, hız ve ivmeye ait türevlerin alınması sonucu oluşan yer değiştirmelerin gerçek zamanlı bir duruma en yakın sonuçların alındığı bir yöntem olduğu gözlemlenmiştir.

Yapılan tez çalışmasında, üçgensel modelleme tekniklerinden yola çıkılarak, tetrahedral geometrik şekil temelinde hacimsel moleküler modelleme yöntemleri geliştirilmiştir. Nesnelerin tetrahedral molekül yapısından yola çıkılarak geliştirilmiş olan potansiyel enerji fonksiyonlarından yola çıkılarak elde edilen kuvvet fonksiyonları ile kütle-yay sistemi ve moleküler modelleme yöntemleri kullanılarak yeni bir yaklaşım geliştirilmiştir.

7. EKLER

Ek- 1. Tetrahedral Geometrik şekle bağlı üçgenlerin karşılaştırılması

```

void ucgenler()
{

    int v,z;

    gecicihafiza();

    // tetgen.txt dosyasından datalar alınıyor...
    TetgenRead(Tetgenfile,uc1,uc2,uc3,uc4);

    // txt dosyalarından datalari okutuyoruz....
    VertexRead(Vertexfile,vertex_x,vertex_y,vertex_z);

    // Geçici hafizadaki dosya koordinat datalari struct bilgisine
    aktariliyor.
    for(i=0;i<Vertexfileline;i++)
    {
        Vkoordinat[i].x=vertex_x[i];
        Vkoordinat[i].y=vertex_y[i];
        Vkoordinat[i].z=vertex_z[i];
    }
    for(i=0;i<Tetgenfileline;i++)
    {
        Uclar[i].n[0]=uc1[i];
        Uclar[i].n[1]=uc2[i];
        Uclar[i].n[2]=uc3[i];
        Uclar[i].n[3]=uc4[i];

        Ucgen[i*4+0].nokta[0]=uc1[i];
        Ucgen[i*4+0].nokta[1]=uc2[i];
        Ucgen[i*4+0].nokta[2]=uc3[i];

        Ucgen[i*4+1].nokta[0]=uc2[i];
        Ucgen[i*4+1].nokta[1]=uc3[i];
        Ucgen[i*4+1].nokta[2]=uc4[i];

        Ucgen[i*4+2].nokta[0]=uc1[i];
        Ucgen[i*4+2].nokta[1]=uc4[i];
        Ucgen[i*4+2].nokta[2]=uc3[i];

        Ucgen[i*4+3].nokta[0]=uc1[i];
        Ucgen[i*4+3].nokta[1]=uc2[i];
        Ucgen[i*4+3].nokta[2]=uc4[i];
    }

    // Ucgen: Tetrahedrona ait sırasız üçgenler...
    // Üçgenleri sıralamak için Ucgen2 degiskeni oluşturduk...
    for(i=0;i<Tetgenfileline*4;i++)
    {

```

```

        Ucgen2[i]=Ucgen[i];
    }
    // Ucgenler Sıralanıyor....
    butun_ucgenleri_sirala(Ucgen2);
    // Üçgen sıralamasını bitirdik....

    // Üçgen karşılaştırmasını yapıyoruz...
    // karşılaştırma için kontrol değişkeni tanımlıyoruz....
    for(i=0;i<Tetgenfileline*4;i++)
    {
        kontrol[i]=0;
    // printf("Kontrol[%d]-> %d \n",i+1,kontrol[i]);
    }
    // Karşılaşma başlıyor....
    // 4 noktayı her defasında tekrar kıyaslamamak için iç içe for döngüsü
    oluşturduk....
    for (v=0;v<Tetgenfileline*4-1;v++)
    {
        for(z=v+1;z<Tetgenfileline*4;z++)
        {
            kontrol[v]=ucgen_karsilastir(Ucgen2[v],Ucgen2[z]);
            if(kontrol[v]==1) break;
        }

        // getch();
    }

}

```

Ek-2. Seçili üçgenlere bağlı normal hesaplama algoritması

```

void Normal_Calculate()
{
    float
ax=0,bx=0,cx=0,ay=0,by=0,cy=0,az=0,bz=0,cz=0,nx=0,ny=0,nz=0;
    float xno=0,yno=0,zno=0,uzunluk=0;

    for(i=0;i<Tetgenfileline*4;i++)
    {
        // bir üçgendeki bir noktanın x koordinatları
        ax=Vkoordinat[Ucgen[i].nokta[0]].x;
        bx=Vkoordinat[Ucgen[i].nokta[1]].x;
        cx=Vkoordinat[Ucgen[i].nokta[2]].x;
        // bir üçgendeki bir noktanın y koordinatları
        ay=Vkoordinat[Ucgen[i].nokta[0]].y;
        by=Vkoordinat[Ucgen[i].nokta[1]].y;
        cy=Vkoordinat[Ucgen[i].nokta[2]].y;
        // bir üçgendeki bir noktanın z koordinatları
        az=Vkoordinat[Ucgen[i].nokta[0]].z;
        bz=Vkoordinat[Ucgen[i].nokta[1]].z;
        cz=Vkoordinat[Ucgen[i].nokta[2]].z ;

        nx = ( (by - ay) * (cz - az) - (bz - az) * (cy - ay) ) ;
        ny = ( (bz - az) * (cx - ax) - (bx - ax) * (cz - az) );
        nz = ( (bx - ax) * (cy - ay) - (by - ay) * (cx - ax) );

        // Normal hesabı için Newell methodunu kullanıyoruz....
        for(p=0;p<3;p++)
        {
            nx += ( by * ( cz - az ) )/3 ;
            ny += ( bz * ( cx - ax ) )/3 ;
            nz += ( bx * ( cy - ay ) )/3;
        }

        uzunluk=sqrt(pow(nx,2)+pow(ny,2)+pow(nz,2));
        xno=nx/uzunluk;
        yno=ny/uzunluk;
        zno=nz/uzunluk;

        if (kontrol[i]==0)
        // benzersiz üçgenlerin normallerini hesaplıyoruz....
        {
            Normal[i].nx=xno;
            Normal[i].ny=yno;
            Normal[i].nz=zno;
        }

        else
        {
            Normal[i].nx=0;
            Normal[i].ny=0;
            Normal[i].nz=0;
        }
    }
}

```


Ek- 3. Tetrahedral bağlantılara bağlı uçların kontrol edilmesi

```

int baglantili_uc()
{
    for(d=0;d<Vertexfileline;d++)
    {
        noktakontrol[d]=-1;
        noktafrekans[d]=0;
        kontrol[d]=0;
    }
    for(d=0;d<Tetgenfileline*4;d++) // Üçgen sayısı kadar...
    {
        for(f=0;f<Vertexfileline;f++) // Nokta sayısı kadar...
        {
            Buc[f]=f+1;
            if (kontrol[d]==0) // tekrar edilmeyen üçgenler...
            {
                if ( Ucgen[d].nokta[0]==Buc[f] || Ucgen[d].nokta[1]==Buc[f] ||
Ucgen[d].nokta[2]==Buc[f] )
                {
                    noktafrekans[f]+=1; // bir noktanın üçgenlerde kaç kez tekrar
edildiğini gösterir...
                    noktakontrol[f]=d; // bağlantılı ucun noktanın kaç nolu
üçgende olduğunu belirtir...
                }
            }
        }

    }

    for(d=0;d<Vertexfileline;d++)
    {
        if (noktafrekans[d]>enbnoktaadetsayisi)
        {
            enbnoktaadetsayisi=noktafrekans[d];
        }
    }
    printf("Enbuyuk Noktafrekans %d \n",enbnoktaadetsayisi);
    return 0;
}

```

Ek- 4. Tetrahedral yüzeylere bağlı yüzey kuvvetlerinin hesaplanması

```

float fyuzeytoplami(yuzeykuvvet *yuzeyf,kenaruzunluk
*rson,kenaruzunluk *rilk,vektorR *Rvekt)
{

    double yf1x=0,yf1y=0,yf1z=0;
    double yf2x=0,yf2y=0,yf2z=0;
    double yf3x=0,yf3y=0,yf3z=0;
    double yf4x=0,yf4y=0,yf4z=0;

    ToplamyuzeykuvvetiX=0;
    ToplamyuzeykuvvetiY=0;
    ToplamyuzeykuvvetiZ=0;

    for(i=0;i<Tetgenfileline;i++)
    {
// 1. yüzeye ait vektörel kuvvet toplami
yf1x=Kb*(rson[i].r[0]-rilk[i].r[0])*Rvekt[i].vRx[0]+Kb*(rson[i].r[1]-
rilk[i].r[1])*Rvekt[i].vRx[1]+Kb*(rson[i].r[5]-
rilk[i].r[5])*Rvekt[i].vRx[5];

yf1x=Kb*(rson[i].r[0]-rilk[i].r[0])*Rvekt[i].vRy[0]+Kb*(rson[i].r[1]-
rilk[i].r[1])*Rvekt[i].vRy[1]+Kb*(rson[i].r[5]-
rilk[i].r[5])*Rvekt[i].vRy[5];

yf1x=Kb*(rson[i].r[0]-rilk[i].r[0])*Rvekt[i].vRz[0]+Kb*(rson[i].r[1]-
rilk[i].r[1])*Rvekt[i].vRz[1]+Kb*(rson[i].r[5]-
rilk[i].r[5])*Rvekt[i].vRz[5];

// 2. yüzeye ait vektörel kuvvet toplami
yf2x=Kb*(rson[i].r[5]-rilk[i].r[5])*Rvekt[i].vRx[5]+Kb*(rson[i].r[2]-
rilk[i].r[2])*Rvekt[i].vRx[2]+Kb*(rson[i].r[3]-
rilk[i].r[3])*Rvekt[i].vRx[3];

yf2y=Kb*(rson[i].r[5]-rilk[i].r[5])*Rvekt[i].vRy[5]+Kb*(rson[i].r[2]-
rilk[i].r[2])*Rvekt[i].vRy[2]+Kb*(rson[i].r[3]-
rilk[i].r[3])*Rvekt[i].vRy[3];

yf2z=Kb*(rson[i].r[5]-rilk[i].r[5])*Rvekt[i].vRz[5]+Kb*(rson[i].r[2]-
rilk[i].r[2])*Rvekt[i].vRz[2]+Kb*(rson[i].r[3]-
rilk[i].r[3])*Rvekt[i].vRz[3];

// 3. yüzeye ait vektörel kuvvet toplami

yf3x=Kb*(rson[i].r[4]-rilk[i].r[4])*Rvekt[i].vRx[4]+Kb*(rson[i].r[3]-
rilk[i].r[3])*Rvekt[i].vRx[3]+Kb*(rson[i].r[1]-
rilk[i].r[1])*Rvekt[i].vRx[1];

yf3y=Kb*(rson[i].r[4]-rilk[i].r[4])*Rvekt[i].vRy[4]+Kb*(rson[i].r[3]-
rilk[i].r[3])*Rvekt[i].vRy[3]+Kb*(rson[i].r[1]-
rilk[i].r[1])*Rvekt[i].vRy[1];

yf3z=Kb*(rson[i].r[4]-rilk[i].r[4])*Rvekt[i].vRz[4]+Kb*(rson[i].r[3]-
rilk[i].r[3])*Rvekt[i].vRz[3]+Kb*(rson[i].r[1]-
rilk[i].r[1])*Rvekt[i].vRz[1];
    }
}

```

```

// 4. yüzeye ait vektörel kuvvet toplami
yf4x=Kb*(rson[i].r[0]-rilk[i].r[0])*Rvekt[i].vRx[0]+Kb*(rson[i].r[2]-
rilk[i].r[2])*Rvekt[i].vRx[2]+Kb*(rson[i].r[4]-
rilk[i].r[4])*Rvekt[i].vRx[4];

yf4y=Kb*(rson[i].r[0]-rilk[i].r[0])*Rvekt[i].vRy[0]+Kb*(rson[i].r[2]-
rilk[i].r[2])*Rvekt[i].vRy[2]+Kb*(rson[i].r[4]-
rilk[i].r[4])*Rvekt[i].vRy[4];

yf4z=Kb*(rson[i].r[0]-rilk[i].r[0])*Rvekt[i].vRz[0]+Kb*(rson[i].r[2]-
rilk[i].r[2])*Rvekt[i].vRz[2]+Kb*(rson[i].r[4]-
rilk[i].r[4])*Rvekt[i].vRz[4];

yuzeyf[i].yfvx[0]=-1*(yf1x+yf2x+yf3x+yf4x);
yuzeyf[i].yfvx[0]=-1*(yf1x+yf2x+yf3x+yf4x);
yuzeyf[i].yfvz[0]=-1*(yf1z+yf2z+yf3z+yf4z);

ToplamyuzeykuvvetiX=ToplamyuzeykuvvetiX+yuzeyf[i].yfvx[0];
ToplamyuzeykuvvetiY=ToplamyuzeykuvvetiY+yuzeyf[i].yfvx[0];
ToplamyuzeykuvvetiZ=ToplamyuzeykuvvetiZ+yuzeyf[i].yfvz[0];

        yf1x=0,yf1y=0,yf1z=0;
        yf2x=0,yf2y=0,yf2z=0;
        yf3x=0,yf3y=0,yf3z=0;
        yf4x=0,yf4y=0,yf4z=0;

    }

    return 1;
}

```

Ek- 5. Tetrahedral yüzeylere bağlı bağ-açı kuvvetlerinin hesaplanması

```

float bagacitoplami(bagacikuvvet *bkuvvet, kenaruzunluk *tet, aci
*kosinus2, aci *kosinus1, vektorR *rvek)
{
    double f1x=0, f1y=0, f1z=0;
    double f2x=0, f2y=0, f2z=0;
    double f3x=0, f3y=0, f3z=0;
    double f4x=0, f4y=0, f4z=0;

    ToplamBagAciKuvvetiX=0;
    ToplamBagAciKuvvetiY=0;
    ToplamBagAciKuvvetiZ=0;

    for(i=0; i<Tetgenfileline; i++)
    {
        // 1. üçgen yüzeye ait vektörel x,y,zler
        f1x= Kteta*(kosinus2[i].kos[0]-
        kosinus1[i].kos[0])*(tet[i].r[4])/(tet[i].r[1]*tet[i].r[3])*rvek[i].vR
        x[4];
        f1y= Kteta*(kosinus2[i].kos[0]-
        kosinus1[i].kos[0])*(tet[i].r[4])/(tet[i].r[1]*tet[i].r[3])*rvek[i].vR
        y[4];
        f1z= Kteta*(kosinus2[i].kos[0]-
        kosinus1[i].kos[0])*(tet[i].r[4])/(tet[i].r[1]*tet[i].r[3])*rvek[i].vR
        z[4];

        // 2. üçgen yüzeye ait vektörel x,y,zler
        f2x= Kteta*(kosinus2[i].kos[1]-
        kosinus1[i].kos[1])*(tet[i].r[3])/(tet[i].r[2]*tet[i].r[4])*rvek[i].vR
        x[3];
        f2y= Kteta*(kosinus2[i].kos[1]-
        kosinus1[i].kos[1])*(tet[i].r[3])/(tet[i].r[2]*tet[i].r[4])*rvek[i].vR
        y[3];
        f2z= Kteta*(kosinus2[i].kos[1]-
        kosinus1[i].kos[1])*(tet[i].r[3])/(tet[i].r[2]*tet[i].r[4])*rvek[i].vR
        z[3];

        // 3. üçgen yüzeye ait vektörel x,y,zler
        f3x= Kteta*(kosinus2[i].kos[2]-
        kosinus1[i].kos[2])*(tet[i].r[2])/(tet[i].r[0]*tet[i].r[1])*rvek[i].vR
        x[2];
        f3y= Kteta*(kosinus2[i].kos[2]-
        kosinus1[i].kos[2])*(tet[i].r[2])/(tet[i].r[0]*tet[i].r[1])*rvek[i].vR
        y[2];
        f3z= Kteta*(kosinus2[i].kos[2]-
        kosinus1[i].kos[2])*(tet[i].r[2])/(tet[i].r[0]*tet[i].r[1])*rvek[i].vR
        z[2];

        // 4. üçgen yüzeye ait vektörel x,y,
        f4x= Kteta*(kosinus2[i].kos[3]-
        kosinus1[i].kos[3])*(tet[i].r[4])/(tet[i].r[1]*tet[i].r[3])*rvek[i].vR
        x[4];
    }
}

```

```

f4y= Kteta*(kosinus2[i].kos[3]-
kosinus1[i].kos[3])*(tet[i].r[4])/(tet[i].r[1]*tet[i].r[3])*rvek[i].vR
y[4];
f4z= Kteta*(kosinus2[i].kos[3]-
kosinus1[i].kos[3])*(tet[i].r[4])/(tet[i].r[1]*tet[i].r[3])*rvek[i].vR
z[4];

    bkuvvet[i].fbagx[0] = -1*(f1x+f2x+f3x+f4x);
    bkuvvet[i].fbagy[0]= -1*(f1y+f2y+f3y+f4y);
    bkuvvet[i].fbagz[0]= -1*(f1z+f2z+f3z+f4z);

    /*printf("bkuvvet[%d].fbagx-> %f\n",i,bkuvvet[i].fbagx[0]);
    printf("bkuvvet[%d].fbagy-> %f\n",i,bkuvvet[i].fbagy[0]);
    printf("bkuvvet[%d].fbagz-> %f\n",i,bkuvvet[i].fbagz[0]);*/

    ToplamBagAciKuvvetiX=ToplamBagAciKuvvetiX+bkuvvet[i].fbagx[0];

    ToplamBagAciKuvvetiY=ToplamBagAciKuvvetiY+bkuvvet[i].fbagy[0];

    ToplamBagAciKuvvetiZ=ToplamBagAciKuvvetiZ+bkuvvet[i].fbagz[0];

    f1x=0,f1y=0,f1z=0;
    f2x=0,f2y=0,f2z=0;
    f3x=0,f3y=0,f3z=0;
    f4x=0,f4y=0,f4z=0;

}
    return 1;
}

```

KAYNAKLAR

- Akarsu V. “Düzlem Üçgende Açıların Kenarlardan Bulunması”, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi* **25**, ISSN 1012-2354 , 2009.
- Akgül M.K., Aydın C., Çarkıt N., Gemci C., Hañer A., Karakoçak K., Öksüz B., Yalvaç E.T., Bilgi Yönetimi, *Türkiye Bilişim Derneği - Kamu Bilişim Platformu VIII- El Kitabı Çalışma Grubu 4*, 2006
- Anonim1 , Nasıl Çalışır – “Bilim, Teknoloji ve İcatlar Ansiklopedisi”, *Gelişim Yayınları*, 3.Cilt Sf 766-768 , 1980
- Anonim 2, Jenko , “Any reliable polygon normal calculation code?”, <http://gamedev.stackexchange.com>, 2011
- Aytekin S., “Sanal Ortamda Cisimlerin Haptic Kol ile Manipölasyonu ve Defarmasyonu”, Yüksek Lisans Tezi, *Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü*, Kütahya, 2011
- Aytekin S. “Sanal Ortamda Nesnelerin Haptic Kol ile Manipölasyonu”, *Uşak Üniversitesi - Akademik Bilişim’12* , 2012
- Bayraktar E., Kaleli F., “Sanal Gerçeklik ve Uygulama Alanları”, *Akademik Bilişim’07, Dumlupınar Üniversitesi, Kütahya*, 2007
- Bedel, O., “Simölasyon Nedir?”, Ansiklopedik Bilgi, [“http://www.femmuhendislik.com/simulasyon/simulasyon-nedir.html”](http://www.femmuhendislik.com/simulasyon/simulasyon-nedir.html), 2009,
- Chen W., JU C., Tang F., Ju C., “Stress distribution of the foot during mid-stance to pushoff in bare foot gait: a 3D finite element analysis.” *Clinical Biomechanics* **16**,614-620., 2001
- Cheung J.T., Zhang M., Leung A.K., Fan Y., “ Three-dimensional finite element analysis of the foot during standing—a material sensitivity study.” *Journal of Biomechanics* **38**, 1045–1054. 2005
- Curless Brian, “A Volumetric Method for Building Complex Models from Range Images ”, “ <http://graphics.stanford.edu/papers/volrange/> ” 1996
- Cutler B., Dorsey J., McMillan L., “Simplification and Improvement of Tetrahedral Models for Simulation”, *Eurographics Symposium on Geometry Processing*, 2001
- Delingette H., “Toward Realistic Soft-Tissue Modeling in Medical Simulation”, *IEEE–0018 – 9219* , 1998
- Edouard J., Robert L., “Computing normals to achieve flat and smooth shading”, <http://www.codeguru.com>, 1999

KAYNAKLAR (Devam Ediyor)

- Erkoç, Ş., “Empricial Many-Body Potential Energy Functions Used In Computer Simulations Of Condensed Matter Properties”, *Middle East Technical University*, Ankara, 1996.3
- Ergin A., Bayraktarkartal E., Ünsan Y., “Sonlu Elemanlar Metodu ve Gemi İnşaatı Sektöründeki Uygulamaları”, *İTÜ Gemi İnşaatı ve Deniz Bilimleri Fakültesi-Türk Loydu Vakfı*, İstanbul, 2000
- Fang Q., Boas D.A., “ Tetrahedral Mesh Generation From Volumetric Binary And Gray-Scale Images”, *ISBI '09 , Sf 1142-1145, USA* ,2009
- Gülçimen B., Özcan R., Ülkü S., “İnsan ayağı biyomekaniğinin sonlu elemanlar yöntemiyle incelenmesi”, *Makine Teknolojileri Elektronik Dergisi* , Cilt:8 No:4, 2011
- Gnu L., “PLY Files an ASCII Polygon Format”
<http://people.sc.fsu.edu/~jburkardt/data/ply/ply.html>, 2007”
- Gözcü, Ş. “Opengl ve Glut ile Oyun Programcılığına Giriş”, *Seçkin Yayınları*, Ankara, 2006
- Hill F.S. , “Modeling Shapes with Polygonal Meshes” , *Interactive Computer Graphics - Fall*, 1999
- Jakobs B., “OpenGL Tutorial – Texture ”, <http://www.videotutorialsrock.com>, 2007
- Kesson M., “Vector Normalizing ”, <http://www.fundza.com/vectors/normalize>, 2002
- Krishnamurthy V., Levoy M., “Fitting Smooth Surfaces to Dense Polygon Meshes”,
<http://graphics.stanford.edu/papers/surfacefitting/>, 1996
- Karl Baur and Elart von Collani: Nuclear Fuel Quality Handbook. Volume I: Fabrication, *Operation, Disposal and Transport of Nuclear Fuel*.
ANT International, Skultuna, Sweden, 2010.
- Land B., “Computing the normal to a surface”, <https://instruct1.cit.cornell.edu>, 1995
- Lawson, C.L., “Generation of Triangular Grid With Application to Contour Plotting”,
California Institute of Technology, Technical Memorandum , 1972

KAYNAKLAR (Devam Ediyor)

- MacDonald J., “Euler’s Method”, *University Of Delaware*,
<http://www.physics.udel.edu/~jim/Ordinary%20Differential%20Equations/Euler%20Method.htm> , 2012
- Mendoza C., Sundaraj K., Laugier C., “Faithful Haptic Feedback in Medical Simulators.”, *International Symposium on Experimental Robotics*, 414-423, 2002
- Mollemans W., Schutyser F., Cleynenbreugel J.V., Suetens P., ”Tetrahedral Mass Spring Model for Fast Soft Tissue Deformation”, *Surgery Simulation and Soft Tissue Modeling International Symposium*, IS4TM 2003 Juan-Les-Pins, France, June 12–13, 2003
- Polatlı A, “Eylemsizlik Prensibi”, Makale, <http://www.makaleler.com>, 2008
- SERWAY R. A., Beichner R. J. , “Newton’un Hareket Kanunları ”, *Palme Yayıncılık*, 2008,
- Scalese Ross J., Obeso Vivian T., Issenberg S. Barry, “Simulation Technology for Skills Training and Competency Assessment in Medical Education”, Journal of General Internal Medicine, Volume 23, Issue 1 Supplement, pp 46-49 , *University of Miami Miller School of Medicine, Miami, USA*, 2008
- Şekercioğlu, A.S. , Duysak, A. , “Application Of Molecular Modeling With Mass-Spring Systems For Computer Simulation and Animation” , *Dumlupınar Üniversitesi, Kütahya*, 2009.
- Şekercioğlu, A.S., “Moleküler Modelleme İle Yumuşak Doku Deformasyonunun Modellenmesi ve Simülasyonu”, Yüksek Lisans Tezi, *Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü*, Kütahya, 2010.
- Trotts I. J., Hamann B., Joy Kenneth I., “Simplification of Tetrahedral Meshes with Error Bounds”, *IEE transactions on visualization and computer graphics*, Vol 5. No:3, 1999
- Turk G., Levoy M, “Zippered Polygon Meshes from Range Images”,
<http://graphics.stanford.edu/papers/zipper/>, 1996
- <http://tr.wikipedia.org>, ”Bilgisayarın Tarihçesi”, *wikipedia.org*, Bilgisayarın Tarihçesi, 2012
- Xu J., Zhaxng J., Huang P., Gu L., Chen F., “Soft tissue modeling using nonlinear mass spring and simplified medial representation” , *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 5083-6, 2007
- West M., "Inner Product", *Game Developer Magazine*, No:32, 2006

KAYNAKLAR (Devam Ediyor)

- Wang Y., Xianfeng G., Thompson P. M., Yau S.T., “3D Harmonic Mapping and Tetrahedral Meshing of Brain Imaging Data”, Paper, ***UCLA, Mathematics Department, Los Angeles, CA, USA - CISE, Univ. of Florida, Gainesville, FL, USA- LONI, UCLA School of Medicine, Los Angeles, CA, USA - Mathematics Department, Harvard University, Cambridge, MA, USA***, 2003
- Weichert F., Schröder A., Landes C., Shamaa A., Awad S.K., Walczak L., Müller H., Wagner M. “Computation of a finite element-conformal tetrahedral mesh approximation for simulated soft tissue deformation using a deformable surface model”, ***International Federation for Medical and Biological Engineering***, 48:597-610, 2010
- Yaşar C., “Plastik Üzerine Kaplamalı Kompozit Malzemelerde Gerilme Analizi”, ***Dokuz Eylül Üniversitesi Mühendislik Fakültesi Makine Mühendisliği Bölümü***, Bitirme Projesi, 2004
- Zhang Y, Wang W, Liang X, Bazilevs Yuri, Hsu MC, Kvamsdal T, Brekken R, Isaksen J. “High-fidelity tetrahedral mesh generation from medical imaging data for fluid-structure interaction analysis of cerebral aneurysms.” ***CMES: Computer Modeling in Engineering & Sciences***, 42(2):131–150, 2009
- Zhendong L. “A mass-spring model for real time cloth deformation ”, ***Multimedia Technology (ICMT), 2011 International Conference***, 2845-2848, 2011

ÖZGEÇMİŞ

Kişisel Bilgiler

ADI SOYADI : Rıdvan YAYLA

DOĞUM YERİ VE TARİHİ : Terme / 19.02.1987

Eğitim Durumu

Lisans Öğrenimi : Balıkesir Üniversitesi Bilgisayar ve Öğr. Tek. Öğrt.

Bildiği Yabancı Diller : İngilizce, Almanca (Temel Seviye)

İş Deneyimi

Çalıştığı Kurumlar :

Balıkesir Polis Meslek Yüksek Okulu (2007 – 2010)

Bilecik Şeyh Edebali Üniversitesi Mühendislik Fakültesi (2010 - ...)

Bilimsel Yayınlar ve Çalışmalar

- Yayla, R., Akçay M., "*Php ve Mysql Tabanlı Öğrenci Bilgi Sistemi Uygulaması*", International Science & Technology Conference, ISTEC 2010, 27-29 October 2010, Famagusta - North Cyprus
- Yayla R., Akçay M., "*Bilecik Üniversitesi Etkinlik ve Duyuru Sistemi*", Akademik Bilişim Konferansı'11, İnönü Üniversitesi, 2-4 Şubat 2011
- Yayla R., "*Bilecik Üniversitesi Moodle Uzaktan Öğretim Sistemi*", Akademik Bilişim Konferansı'12, Uşak Üniversitesi, 1-3 Şubat, 2012

İletişim

Adres: Bilecik Şeyh Edebali Üniversitesi Mühendislik Fakültesi 11210

Gülümbe Kampüsü Bilecik / Türkiye

Tel: 0 228 214 12 28 - E-Posta: ridvan.yayla@bilecik.edu.tr

