

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

93701

KALIP TASARIMINDAKİ SORUNLARIN MATEMATİKSEL
YAKLAŞIMLARLA İNCELENMESİ

Makina Mühendisi Tayfun YAMAN

F.B.E. Makina Mühendisliği Anabilim Dalı Konstrüksiyon Programında

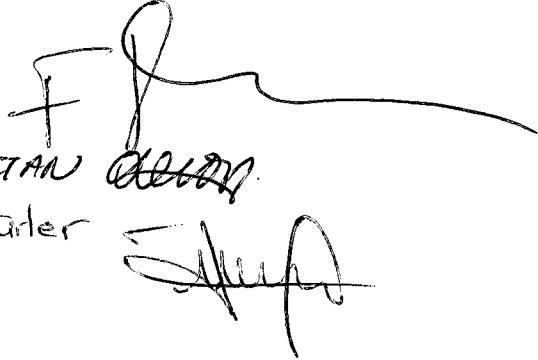
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Doç. Dr. Ferhat DİKMEN

Prof. Dr. Erhan ALTAN

Doç. Dr. Mesut Özgürler



İSTANBUL, 2000

İÇİNDEKİLER

Sayfa

ŞEKİL LİSTESİ	i
ÖNSÖZ	iii
ÖZET	iv
ABSTRACT	v
1. GİRİŞ	1
2. BİLGİSAYAR DESTEKLİ TASARIM, İMALAT VE ANALİZ	2
2.1 Bilgisayar Destekli Tasarım (CAD).....	2
2.2 Bilgisayar Destekli İmalat (CAM).....	7
2.3 Bilgisayar Destekli Analiz (CAE)	15
2.3.1 Sonlu elemanlar analizi teorisi.....	15
2.3.1.1 Direkt metod.....	16
2.3.1.2 Varyasyonel formülasyon.....	18
2.3.1.3 Ağırlıklı kalıntılar yöntemi.....	22
2.3.1.4 Gerilmelerin hesabı.....	25
2.3.2 Bilgisayar destekli sonlu elemanlar analizi.....	25
2.3.2.1 Kullanılacak programın (ANSYS) tanıtılması.....	26
2.3.2.2. Mührelerin sonlu elemanlar analizlerinin yapılması.....	30
3. SONUÇLAR ve ÖNERİLER	43
3.1 Analiz Sonuçlarının Sunumu Ve Sonuçların İrdelenmesi.....	43
3.2 CAD/CAM/CAE Kavramları Üzerine Yorum ve Öneriler.....	48
KAYNAKLAR	49
EKLER	50
Ek 1 Örnek Kalıp Projesinin Detay Resimleri.....	51
Ek 2 Örnek Analizde Kullanılan ELEMEN 92'nin özellikleri	63
Ek 3 Gaus Eliminasyon Metodunun FORTRAN Programlama Dili İle Yazılmış Kodları	66
Ek 4 C++ Programlama Dili İle Yazılmış Sonlu Elemanlar Analiz Programı Kodları	69
ÖZGEÇMİŞ	83

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1	Dişi kesici parçasının temel geometrisinin oluşturulması aşaması	3
Şekil 2.2	Oluşturulan temel geometriye kalınlık verilmesi aşaması	4
Şekil 2.3	Katı model üzerinde nihai işlemlerin yapılması	5
Şekil 2.4	Altparçaların birleştirilerek kalıp montajının elde edilmesi işlemi	6
Şekil 2.5	Model teknik resimlerinin otomatik olarak çizdirilmesi aşaması	7
Şekil 2.6	CAM programına genel işleme tipinin belirtilmesi	9
Şekil 2.7	Kesici takımın seçilmesi	10
Şekil 2.8	Takım hareket sınırlarının belirtilmesi	11
Şekil 2.9	İşleme stratejisinin belirtilmesi	12
Şekil 2.10	İşleme sonucu program tarafından verilen rapor	13
Şekil 2.11	Elde edilen takım yollarının CNC kodlarına dönüştürülmesi	13
Şekil 2.12	Post-Prosesör tarafından dönüştürülen CNC kodları	14
Şekil 2.13	Konik çubuk	16
Şekil 2.14	Konik çubuğun basitleştirilmiş modeli	17
Şekil 2.15	İki uçtan mesnetlenmiş lineer elastik çubuk	20
Şekil 2.16	ANSYS 5.4 arayüzünün genel görünümü	27
Şekil 2.17	Preprosesör menüsünün içerdiği alt menüler	28
Şekil 2.18	Çözüm menüsünün içerdiği alt menüler	28
Şekil 2.19	Postprosesör menüsünün içerdiği alt menüler	29
Şekil 2.20	Analizi yapılacak olan mührenin geometrisi ve ölçüleri	30
Şekil 2.21	Geometrinin ANSYS ortamına IGES formatı kullanılarak transfer edilmesi	31
Şekil 2.22	ANSYS tarafından katı modele dönüştürülmüş olan geometri	32
Şekil 2.23	Eleman tipinin seçilmesi aşaması	33
Şekil 2.24	Malzeme özelliklerinin belirlenmesi işlemi	33
Şekil 2.25	Sonlu elemanlara bölme özelliklerinin belirlenmesi	34
Şekil 2.26	Geometrinin sonlu elemanlara bölünmesi işlemi	35
Şekil 2.27	Kritik bölgelerin ağ yapısında iyileştirme işleminin yapılması	36
Şekil 2.28	Rafine işlemi yapılmış geometri ve ağ yapısı	36
Şekil 2.29	Düğüm noktalara yüklerin uygulanması	37
Şekil 2.30	Sınırlamaların belirlenmesi işlemi	38
Şekil 2.31	Eleman matrislerinin oluşturulması aşaması	39
Şekil 2.32	Eleman matrislerinin birleştirilerek global matrisin eldesi aşaması	40
Şekil 2.33	Lineer cebrik denklem sisteminin çözümlenmesi aşaması	40

Şekil 2.34	Yerel yerdeğiřtirme deęerlerinden gerilme ve toplam deformasyon hesabı ařaması	41
Şekil 3.1	Küçük dörtgen mührenin toplam deformasyon durumu	44
Şekil 3.2	Küçük dörtgen mührenin toplam gerilme durumu	44
Şekil 3.3	Büyük dörtgen mühre toplam deformasyon durumu	45
Şekil 3.4	Büyük dörtgen mührenin toplam gerilme durumu	45
Şekil 3.5	Büyük altıgen mühre toplam deformasyon durumu	46
Şekil 3.6	Büyük altıgen mührenin toplam gerilme durumu	46
Şekil 3.7	Küçük altıgen mühre toplam deformasyon durumu	47
Şekil 3.8	Küçük altıgen mühre toplam gerilme durumu	47
Şekil Ek 1.1	Solid 92 eleman yapısı	51



ÖNSÖZ

Bu yüksek lisans tezinde, endüstriyel alanlarda her geçen gün daha fazla önem kazanan bilgisayar destekli tasarım (CAD), bilgisayar destekli imalat (CAM) ve bilgisayar destekli analiz (CAE) kavramlarının tanımlarını yapılmış, bu kavramlar hakkında teorik bilgiler verilmiş ve pratik uygulamaları hakkında bilgi verebilmek için örnek bir kalıp projesi yapılmıştır. Özellikle bilgisayar destekli analiz üzerinde durulmuş, yapılan kalıp projesine ait kritik parçaların Sonlu Elemanlar Analizleri yapılmış ve C++ programlama dili kullanılarak sonlu elemanlar analizi yapan bir bilgisayar programı da ilave edilmiştir.

Bu çalışmalar katkılarından dolayı, benden yardımlarını esirgemeyen ve tez danışmanlığımı yapan sayın hocam Doç. Dr. Ferhat DİKMEN'e, yine her zaman fikirleri ile bana yol gösteren sayın hocam Doç. Dr. Muharrem BOĞOÇLU'ya, pratik uygulamalarda yardımlarını esirgemeyen iş arkadaşlarım Bosal-Mimaysan A.Ş. çalışanlarına ve tez çalışmalarımnda kullandığım bilgisayar programlarında bana destek veren değerli arkadaşım MTM Bilgi Teknolojileri sahibi Mustafa TİĞREK'e teşekkürü borç bilirim.

ÖZET

Çağımıza damgasını vuran bilgisayar sistemleri, makina – kalıp endüstrisine de CAD/CAM/CAE kavramlarını sokmuş ve vazgeçilemez bir unsur olarak yer edinmiştir. Bu durum çok büyük bir CAD/CAM/CAE pazarı oluşturmuş ve çok çeşitli ürünler pazarda pay sahibi olabilmek için büyük rekabet içine girmişlerdir. Bu rekabet ortamı içerisinde kullanıcı ihtiyaçlarına cevap veren, uygun fiyatlı CAD/CAM/CAE sisteminin seçilmesi ve yatırımların seçilen sistem üzerine yapılması, bu kavramlar hakkında çok iyi bir teknik bilgiye sahip olunmasını gerektirmektedir. Özellikle, ülkemizde diğer kavramlara göre daha yeni olan Sonlu Elemanlar Analizi kavramı endüstriyel uygulamalarda CAE programları ile uygulama alanı bulmaya başlamıştır. Ancak Sonlu Elemanlar Analizi hakkında yeterli sayıda Türkçe yayının bulunmaması ve endüstride bu konuda bilgi kullanıcının az olması nedeni ile bu tip programlar ülkemizde uygun ve etkin bir biçimde kullanılamamaktadır. Bu nedenle tez içerisinde CAD/CAM kavramları örnek bir kalıp projesi yapılarak aşamalar halinde açıklanmıştır. Ancak daha çok CAE ve Sonlu Elemanlar Analizi üzerinde durulmuştur. Sonlu Elemanlar Teorisi açıklanmış, ve tasarımı yapılan kalıbın kritik parçalarının analizi yapılarak analiz sonuçları irdelenmiştir. Ayrıca tez çalışmasının sonunda, Sonlu Elemanlar Analizi prosedürlerinin bilgisayara nasıl uyarlandığının görülebilmesi için C++ programlama dili ile yazılmış kodlar, ekler kısmına ilave edilmiştir. Tez çalışması sonucu, CAD uygulamalarında Katı Modelleme tekniği ile yapılan tasarımların, teknik dokümanların hazırlanmasında, CAM aşamasında ve CAE aşamasında uygulama kolaylıkları sağladığı ancak yüzey modellemenin de analiz aşamasında zaman tasarrufu sağladığı, CAD sistemine entegre çalışan CAM sistemlerinin tercih edilmesi durumunda veri kaybının önlenmesi ve zaman tasarrufunun sağlandığı, Sonlu Elemanlar Analizi aşamasında ise ince parçalar için kabuk elemanların zaman tasarrufu sağlaması nedeni ile daha kullanışlı olduğu sonuçlarına varılmıştır.

Anahtar Kelimeler: CAD, CAM, CAE, Sonlu Elemanlar Analizi

ABSTRACT

The CAD/CAM/CAE concepts have taken their places as a must in machine-tool industry by the great development of computer systems. As a result of this development, a competitive market has been constituted and a great variety of these products have taken place in market. Choise of a usable and rantable CAD/CAM/CAE system and make investments on the chosen system, needs a good technical background about CAD/CAM/CAE systems. Lack of Finite Element Analysis publications in Turkish and technical knowledge of the theory causes inappropriate usage of CAE programs. For this reason, the CAD/CAM procedures are explained step by step in a sample stamping tool project. CAE and Finite Element Analysis concepts are taken as the main objectives in the project. The theory of Finite Element Analysis is explained briefly and the analysis of the critical parts in the project are made by an analysis software. To describe the usage of finite element procedures in a software, the program codes of a Finite Element Analysis program written in C++ language are placed at the end of the thesis. According to the conclusions of this thesis study, working with solid models in CAD modelling brings advantages in documentation, manufacturing and analysing stages of a project bu this method takes longer than working with surfaces. The choise of integrated CAD/CAM systems rather than interfaced CAM systems bring the safety of data loss. The usage of shell elements for thin parts in Finite Elements Analysis results in faster analysis performance rather than solid elements usage.

Keywords: CAD, CAM, CAE, Finite Elements Analysis

1. GİRİŞ

Bir ülkenin uluslararası alanda söz sahibi olabilmesi ve ekonomik refahı, o ülkenin sanayiinin gelişmişliğine bağlıdır. Ülkemiz de bu alanda gün geçtikçe büyük mesafeler katetmektedir. Ancak güçlü bir sanayiinin hayatını devam ettirebilmesi için teknolojik yenilikleri sürekli olarak takip etmesi, uygulayabilmesi ve kendi teknolojisini geliştirmesi gerekmektedir. Ülkemizde de bu doğal teknolojik gelişim gözlenmektedir. Özellikle CNC tezgahları dolayısıyla da CAD / CAM / CAE uygulamalarının ülkemize yerleşmesi ile bu gelişim daha da çarpıcı bir hal almaktadır. Bu gelişim özellikle kalıp imalat sanayiinde kendini göstermektedir.

Ancak kalıp tasarım ve imalatı esnasında, etkiyen kuvvetleri çok karmaşık olması nedeniyle, malzeme boyutları tespitinde hala tecrübeye dayalı yöntemler kullanılmaktadır. İşte Sonlu Elemanlar bu nedenle ülkemizde kendisine hızla uygulama alanı bulması gereken bir yöntem olmaktadır. Kalıp tasarımı esnasında, çalışma anında etkiyecek kuvvetlerin ve bu kuvvetlerin kalıp üzerinde meydana getireceği gerilmelerin hesaplanması sonucu, hem daha emniyetli hem de daha ekonomik kalıplar tasarlanabilecektir.

Bu yüksek lisans tez çalışmasındaki amaç, bilgisayar destekli araçların kullanılması vasıtası ile bir delik kalıbının tasarım, imalat ve analizinin yapılması, bu çalışmaların aşama aşama anlatılması, CAD (bilgisayar Destekli Tasarım), CAM (Bilgisayar Destekli İmalat) ve CAE (Bilgisayar Destekli Analiz) aşamalarında önemli kriterlerin belirlenmesi, çalışma tekniklerinin, ve programların yazılım mimarilerinin incelenmesi ve sonuç olarak da bu incelemelere dayalı olarak uygun teknik ve program mimarilerinin belirlenmesidir. Ayrıca endüstride çok yaygın olarak uygulama alanı bulamamış olan Sonlu Elemanlar Analizi kavramı üzerinde durulmuş, teorik bilgiler verilmiş, örnek projenin kritik elemanlarının mekanik analizleri yapılmış ve Sonlu Elemanlar Analizi prosedürlerinin, bilgisayar kodlarına nasıl dönüştürülebileceğinin açıklanabilmesi için örnek bir analiz programı C++ bilgisayar programlama dili kullanılarak yazılmış, kodlar tezin ekler kısmında verilmiştir.

2. BİLGİSAYAR DESTEKLİ TASARIM, İMALAT VE ANALİZ

2.1 Bilgisayar Destekli Tasarım (CAD)

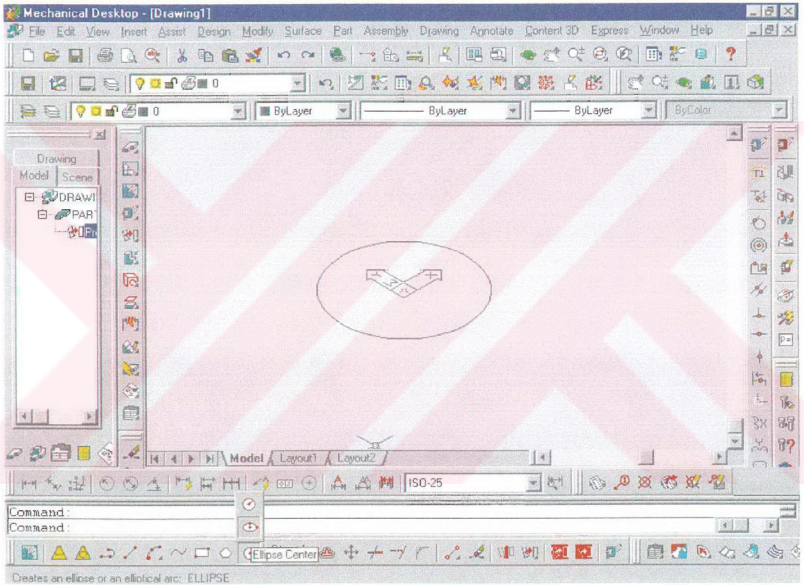
Bilgisayar ortamında tasarım (CAD), diğerlerine göre ülkemizde en eski kullanım alanına sahip olan kavramdır. Ancak en önemli olanıdır. Zira imalat tasarımıyla başlamaktadır ve günümüzde bu amaca hizmet eden bilgisayar programları çok büyük bir gelişim göstermişlerdir. Artık imalatı yapılacak olan parçalar, iki boyutlu çizgiler halinde çizilmemekte, katı modeller olarak modellenmekte ve sanal ortamda (ağırlık, atalet momentleri, asal eksenleri vs.) tüm özellikleri ile görülebilmektedir. Bu sayede tasarımı yapılan parçaların ağırlık hesaplarından gerilme analizlerine kadar tüm çalışmalar bu modeller üzerinde yapılabilmektedir. Eğer bir kalıp tasarımı söz konusu ise, kalıbı meydana getiren parçalar ayrı ayrı dosyalar halinde tek tek tasarlanmakta, modellenmekte ve daha sonra başka bir dosyada birleştirilmektedir. Bu sayede çok karmaşık kalıpların tasarımı dahi basitleştirilebilmektedir. Tasarımı biten ve montajı bilgisayar ortamında tamamlanmış bir kalıbın parça teknik resimlerinin ve montaj teknik resimlerinin çıkartılması, boyutlandırma işlemleri, antet hazırlanması gibi işlemler otomatik olarak yaptırılabilir.

Bilgisayar ortamında tasarım amacına hizmet eden birçok bilgisayar programı bulunmaktadır. Bunlardan bazılarını örnek vermek gerekirse Autodesk Mechanical Desktop, Pro-Engineer, Unigraphics, Ideas, Catia, Solidworks, Cimatron programları sayılabilir. Bu programların bazıları tasarım (CAD) işlemlerinin yanında imalat işlemlerini de (CAM) yapmaktadır. Bu tip programlar entegre mimariye (integrated) sahiptir. Sadece CAD amaçlı kullanılan programlarda ise CAM işlemleri, bu programlara entegre edilebilen ayrı CAM programları ile yapılabilir. Bu programların birbirinden farkı genelde komut isimleri veya işlem sıralarında oluşmaktadır. Ancak temel işlem adımları ve prensipleri genelde birbiri ile aynıdır. Bu nedenle bir CAD programının iyi bir şekilde kullanılabilmesi, diğer programlar için de potansiyel bir kullanıcı durumuna gelinmesi anlamını taşımaktadır. Burada bir CAD programında parça modelinin oluşturulması için gerekli işlem adımları açıklanırken Autodesk Mechanical Desktop R4 programından faydalanılacaktır. Mechanical Desktop R4 programı, daha temel işlemler için kullanılan Autocad programı üzerine ADSRX makroları yazılarak elde edilmiş, oldukça kabiliyetli ve opsiyonu epeyce fazla olan bir tasarım programıdır. Hem yüzeylerle hem de katılarla çalışabilmekte, otomatik olarak teknik resim çıkarabilmekte ve entegre olarak Sonlu Elemanlar Analizi paketi de içermektedir. Kullanım açısından da oldukça kolay ve pratiktir. Parça modellemede yapılacak işlem adımlarının açıklanması için bu programdan faydalanılmasının nedenlerinden biri de,

ülkemizde Autocad temelinin diğerlerine göre daha yaygın olması ve bu nedenle işlem adımlarının daha kolay anlaşılabilmesidir.

Genel olarak bir parçanın bilgisayar ortamında katı modelinin yapılması aşağıdaki aşamalardan oluşmaktadır;

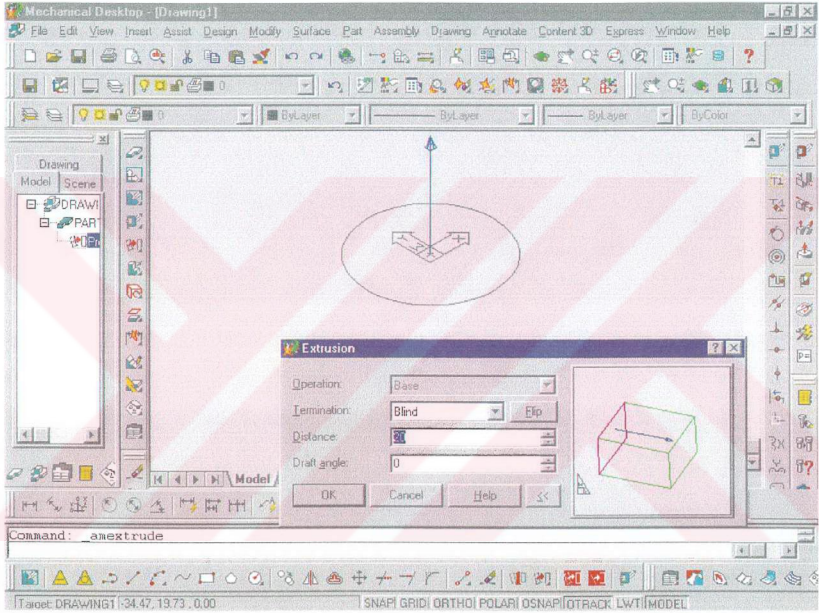
1. Modeli yapılacak olan parça temel geometrisinin 2 boyutlu olarak çizilmesi (bkz. Şekil 2.1). Bu aşamada genelde tasarlanacak olan geometrinin kaba sınırı çizilir.



Şekil 2.1 Dişi kesici parçasının temel geometrisinin oluşturulması aşaması.

2. Oluşturulan temel geometriye kalınlık verilmesi veya bu geometrinin bir yörunge boyunca süpürülerek temel katı modelin elde edilmesi (bkz. Şekil 2.2). Bu aşamada kullanılabilecek teknikler genelde tüm CAD sistemlerinde aynıdır. Bu teknikler genelde kapalı bir kesit alanına kalınlık verme (extrude), kapalı bir kesit alanını bir eksen etrafında yuvarlama (revolve), kapalı bir kesit alanın bir patika boyunca süpürülmesi (sweep) ya da iki kapalı kesit alanın aradaki bir patika boyunca süpürülerek birbirine bağlanması (loft) şeklindedir. Bu tekniklerin seçimi,

tasarlanmak istenen geometrinin şekil özelliklerine bağlı olarak değişmektedir. Dönel simetrik parçalarda eksen etrafında döndürme ile model elde etme kullanışlı iken iki farklı geometriye sahip kesitlerin birleştirilebilmesi için loft işlemi tek çözümdür. Bir CAD sisteminin modelleme gücünü ortaya koyan en önemli aşamalardan biri de budur. CAD sistemlerinin modelleme güçleri, barındırdıkları geometri oluşturma teknik sayısı ile doğru orantılıdır.

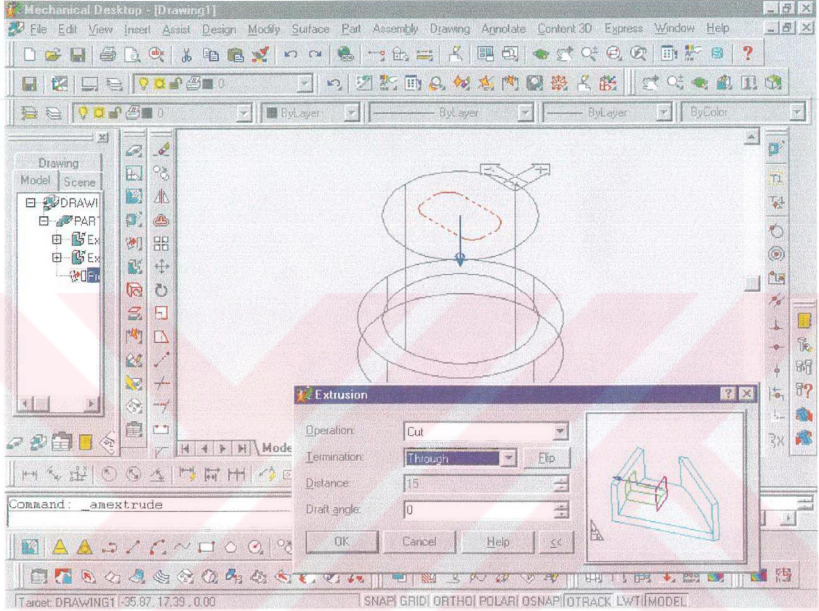


Şekil 2.2 Oluşturulan temel geometriye kalınlık verilmesi aşaması.

3. Elde edilen temel katı modelin üzerinde diğer işlemlerin yapılması ile (pah kırma, delik delme, yuvarlatma gibi) parça gerçek modelinin elde edilmesi (bkz. Şekil 2.3). Bu aşamada parçanın üzerindeki detaylar yapılmaktadır.

Şayet bir kalıp gibi alt elemanlardan oluşan bir sistem tasarlanıyorsa, bu sistemi oluşturan parçalar, yukarıdaki aşamalardan geçilerek tek tek elde edilir ve bu parçaların tümü başka bir dosyada monte edilir. Bu işlem için montajın yapılacağı dosya açılır ve bu dosya içinde bir altparça kataloğu oluşturulur. Bu altparçalar katalogdan tek tek seçilerek resme dahil edilir. Bu

parçalar resme dahil edilirken, parçalar üzerinde kısıtlamalar yapılarak serbestlik dereceleri belirtilir (bkz. Şekil 2.4).

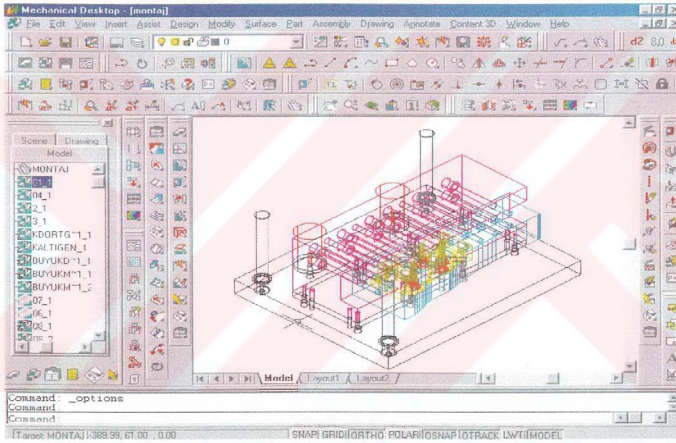


Şekil 2.3 Katı model üzerinde nihai işlemlerin yapılması

Son olarak montajı da tamamlanmış olan bir katı modelin teknik resminin otomatik olarak çizdirilmesi aşaması yapılacaktır (bkz Şekil 2.5). Teknik resmi de çizilmiş olan kalıp modelinin tasarım aşaması da tamamlanmış sayılmaktadır.

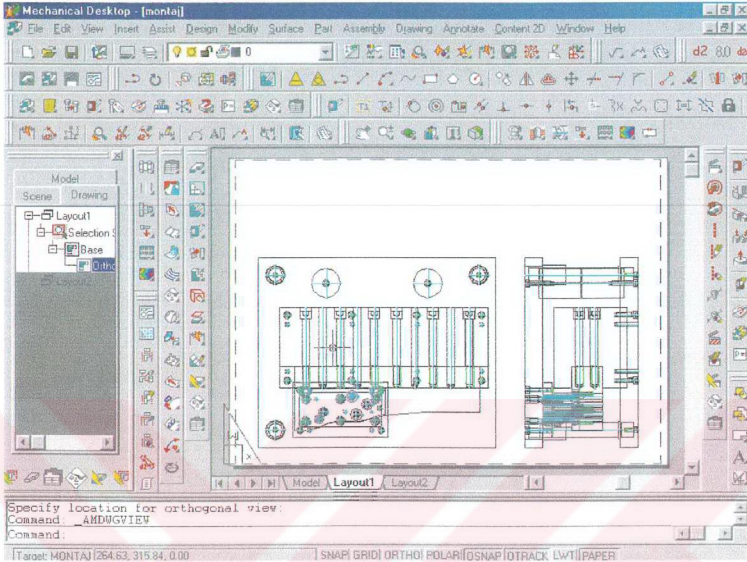
Katı Modelleme ve Yüzey modelleme olmak üzere iki parça modelleme tekniği mevcuttur. Bu örnek projede kalıp alt parçalarının tasarımı katı modeller şeklinde yapılmıştır. Bu sayede her bir parçanın ağırlık, malzeme yoğunluğu gibi özellikleri bu parçalara verilebilmekte ve bu basit hesaplamalar CAD programında yaptırılabilir. Bu nedenle katı modelleme tekniği gün geçtikçe daha çok tercih edilmekte ve uygulama alanı bulmaktadır. Yüzey modelleme tekniği ise kalınlığı olmayan yüzeyler yardımı ile parça modelleme tekniğidir. Katı modellemeye göre daha

eski bir tekniktir. Ancak bazı karmaşık parça yüzey şekillerinde katı modelleme tekniği yetersiz kalabilmekte ve yüzey modelleme tekniği kullanımı zorunlu olabilmektedir. Ayrıca özellikle saç parçalar gibi kalınlığı diğer boyutlarına göre daha ince olan parçalarda yüzey modelleme daha hızlı olabilmekte ve bu sayede zamandan tasarruf edilmektedir. Gelişmiş CAD sistemlerinde her iki metotla da çalışmak mümkün olabilmektedir. Artık çoğu CAD sistemi yüzeylerle oluşturulmuş modelleri, kapalı yüzeylerin katıya dönüştürülmesi ve/veya yüzeye kalınlık verilerek katı model elde edilebilmesi seçenekleri sağlamaktadır. Bu özellik hibrit çalışma özelliği olarak adlandırılmakta ve CAD sistemlerinin seçimi için en önemli kriterlerden biri olarak karşımıza çıkmaktadır.



Şekil 2.4 Altparçaların birleştirilerek kalıp montajının elde edilmesi işlemi.

CAD sistemlerinin seçiminde belirleyici özelliklerden biri de parametrik çalışma özelliklerinin olup olmamasıdır. Parametrik çalışabilme özelliği, bir katı veya yüzey modelinin yapıldıktan sonra, parçayı silip yeniden yapmadan, temel özelliklerini (temel geometri, detay geometri) değiştirebilme kabiliyetidir. Bu kabiliyet, tasarımda değişiklik gerektiği durumlarda zaman tasarrufu ve basit kullanım özellikleri sağlamaktadır. CAD sistemlerinin seçimi için en önemli kriterlerden biri olarak karşımıza çıkmaktadır.



Şekil 2.5 Model teknik resimlerinin otomatik olarak çizdirilmesi aşaması.

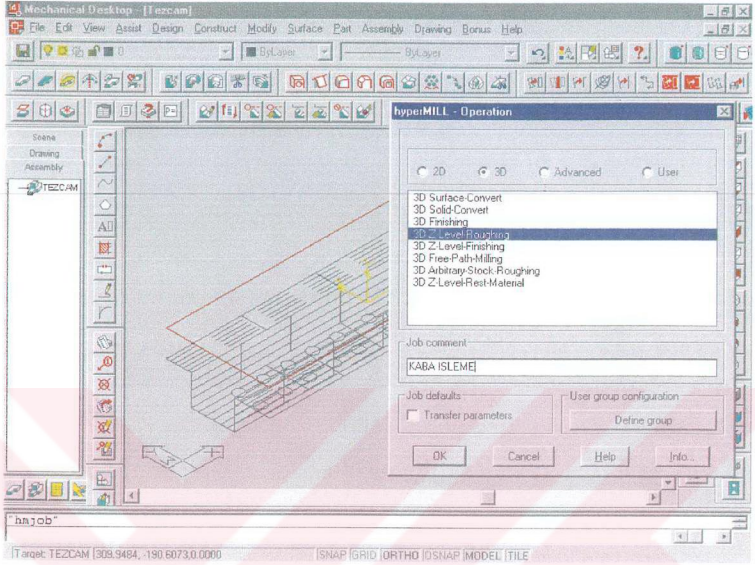
Bir CAD sisteminin, modelleme kabiliyeti dışında seçiminde kriter olabilecek diğer bir özellik ise, diğer CAD ve CAM sistemleri ile veri alış verişi kabiliyetidir. Farklı CAD sistemleri arasında veri alış verişi, “transfer formatları” olarak anılan dosya tipleri ile sağlanmaktadır. Bu dosya tiplerinden yıllardır en popüler olanı IGES formatındaki dosyalardır. Ancak IGES formatındaki dosyalar, transfer esnasında katı modelleri yüzey modeller haline dönüştürmektedir. Son yıllarda bu formata alternatif olarak, bünyesinde katı modeli saklayabilen ve katı model transferi sağlayabilen ACIS (SAT) dosyaları kullanılmaktadır. Diğer transfer formatları ise genel olarak STL, DXF, VDA ve VRML olarak sıralanabilir.

2.2 Bilgisayar Destekli İmalat (CAM)

Bir kalıbın bilgisayar ortamında tasarımının tamamlanması ardından bu kalıbı oluşturan parçaların imalatının yapılabilmesi için CNC kodlarının çıkarılması gerekmektedir. Bu kodların çıkarılabilmesi için bir CAM programından faydalanılmalıdır. Bir CAM programı kullanılarak CNC işleme merkezine işleme kodlarının yollanması işlemi şu aşamalardan oluşmaktadır;

1. Takım yolları çıkarılacak olan parçanın geometrisinin CAM programına tanıtılması işlemi ilk işlemdir. Eğer kullanılan CAD programı ile entegre olmayan harici bir CAM programı kullanılıyorsa, geometrilerin CAM programına tanıtılması işlemi kullanıcı tarafından yapılmalıdır. Bunun için tasarımcı, tasarladığı parçaların geometrilerini ortak bir dosya formatında (DXF, IGES, VRML, STL gibi) CAD programı içerisinde dışarı kaydetmelidir (export işlemi). Bu formatlar genel haberleşme formatı olarak kabul edilmekte ve genelde (özellikle IGES) tüm CAM programları bu formatları desteklemektedir. Daha sonra bu formatta kaydedilmiş olan dosya CAM programına okutulur ve geometri nakledilmiş olur. Ancak CAD programına entegre edilmiş bir CAM programı kullanılıyorsa bu işleme gerek kalmayacak ve Cam programı geometriyi kendiliğinden tanıyacaktır. Bu nedenle CAD sistemine entegre olarak çalışan CAM sistemlerinde veri kaybı olasılığı yok edilmekte ve dosya dönüştürme işlemi yapılmayacağından zaman tasarrufu sağlanmaktadır.

2. Tanıtma işlemi sonrasında ise yapılacak olan işlem, genel işleme tipinin belirtilmesidir. Genel işleme tipleri genelde kaba işleme ya da hassas işleme (finishing) olarak ikiye ayrılabilir (bkz Şekil 2.6). Bir CAM programı ile imalat yapabilmek için kaba işleme ve hassas işleme modüllerinin olması zorunludur. Ancak CAM programları arasında seçim yapabilmek için bunlara ek olarak kullanıcıya sunulan işleme seçenekler, söz konusu CAM programının hanesine artı puan eklemektedir. Genelde ek seçenekler arasında döküm paylı işleme seçeneği, otomatik takım kalıntılarını işleme seçenekleri ve sabit yüzey pürüzlülüğü sağlayarak işleme seçenekleri benzeri opsiyonlara rastlanmaktadır. Ayrıca bir CAM programı yüzey ve katı modeller gibi üç boyutlu geometrilerle çalışabildiği gibi iki boyutlu çizgilerden oluşan kontürlerle de işleme yapabilmektedir. Zira iki boyutlu geometrilerle kalıp endüstrisinde birçok işlem (delik delme, plaka kenarlarını işleme, havuz açma gibi) pratik ve hata payı minimuma indirilerek yapılabilir. Ayrıca iki boyutlu kontürlerle yapılan işlem zamanları, üç boyutlu modellerin işleme zamanlarına göre çok daha kısa sürmektedir. Bu nedenle mümkün olan durumlarda iki boyutlu işleme stratejileri tercih edilmelidir.

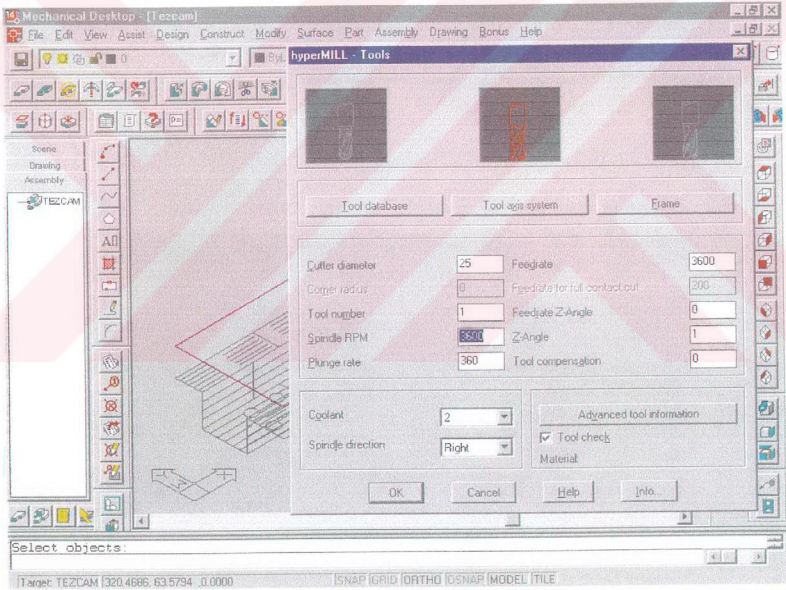


Şekil 2.6 CAM programına genel işleme tipinin belirlenmesi.

3. Genel işleme stratejisinin belirlenmesinin ardından işlemenin yapılmasında kullanılacak takım tipi seçilecektir. Takım tipinin belirlenmesi esnasında takım şekli (küre takım, HSS veya Sert Metal takım, Düz kenarlı takım gibi) seçilir, takım devir sayısı ve devir başına ilerleme miktarları, takım üreticisinin katalogundan alınan bilgiler doğrultusunda işlenecek malzemeye göre belirlenir (bkz Şekil 2.7).

4. Takım tipinin ve hareket hızlarının belirlenmesi işlemi ardından işleme esnasında takımın yapabileceği hareket sınırları belirlenmelidir. Bu işlem, çalışma, makina ve iş parçası güvenliği bakımından en önemli aşamadır. Bu aşamada takımın minimum inebileceği yükseklik değeri, hızlı hareketlerini yapabileceği minimum yükseklik değeri gibi sınırlar belirlenmektedir. Takımın inebileceği minimum yükseklik değeri, iş parçasının taban değeri ile, takım hızlı hareketlerinin minimum yükseklik değeri ise iş parçası tepe değeri ile sınırlıdır. Burada Cam programı kullanıcı ile etkileşim halinde olmalıdır (bkz. Şekil 2.8). Birçok program verilen sınırların durumunu görsel olarak kullanıcıya belirtmekte ve tehlikeli sınır değerleri verilmesi durumunda kullanıcıyı uyarmaktadır. Bu aşamada yapılacak olan bir hata, iş parçası güvenliğini, işçi güvenliğini ve

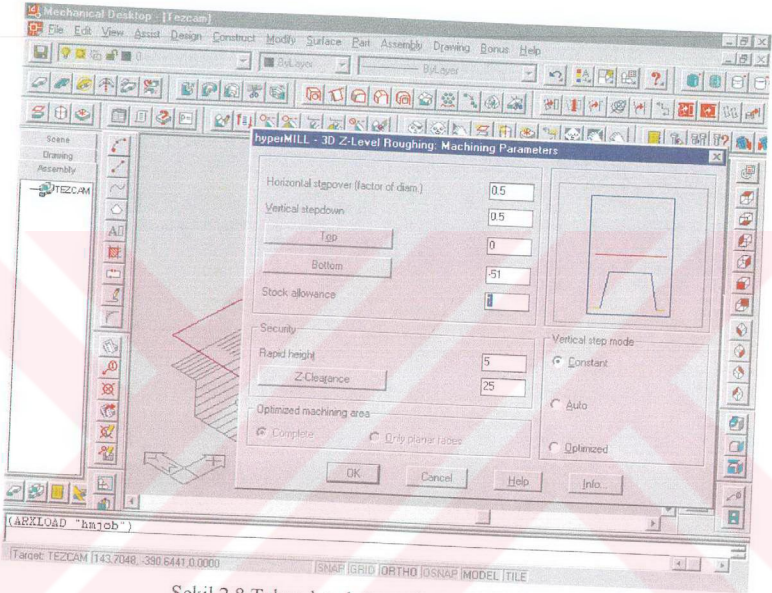
makina güvenliğini doğrudan etkilediği için, CAM sistemlerinin seçiminde çok önemli bir yer tutmaktadır. Minimum inme sınırlarının, iş parçası taban sınırından daha aşağıya verilmesi durumunda, işlenen yüzeylerde boşluklar varsa “dalma” olarak tabir edilen, takımın iş parçasına saplanma olayı meydana gelmektedir. Yeni teknolojiye sahip gelişmiş programların birçoğunda çarpma kontrollü işleme tekniği kullanıldığından, dalma olayının meydana gelme ihtimali minimuma indirilmektedir. Bu teknikte yüzey işleme esnasında bir boşlukla karşılaşılması durumunda program takım çapı ile boşluk büyüklüğünü karşılaştırır. Bu karşılaştırma sonucunda, takım iş parçasına deliğe girebilecek kadar sokulur diğer yüzeyler bozulmaz. Bu sayede yüzey geometrisindeki, kullanıcı tarafından farkedilemeyen çok küçük boşluk değerleri, dalma olayı ile sonuçlanmaz ve program güvenli bir biçimde çalışır.



Şekil 2.7 Kesici takımın seçilmesi .

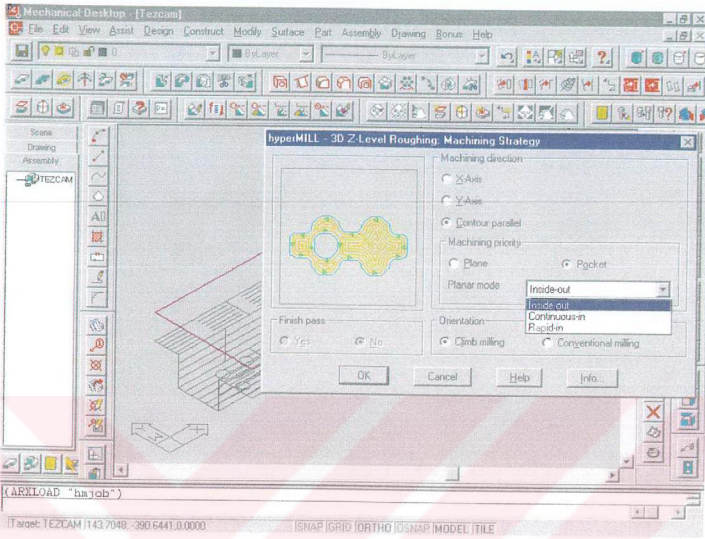
5. İşleme sınırlarının belirlenmesi ardından sıra geometrinin detay işleme stratejisini belirleme aşamasına gelmiştir (bkz. Şekil 2.9). Bu aşamada belirlenen strateji, elde edilen iş parçası yüzey kalitesine ve işleme zamanına etki etmektedir. Burada amaç takımın maksimum noktalara kalkma

sayısı minimum olacak ve en temiz yüzey kalitesini elde etmemizi sağlayacak işleme stratejisinin belirlenmesidir. Minimum inme kalkma sayısını veren işleme stratejisi, bilgisayar ortamında deneme yanılma metodu izlenerek elde edilebilir. Ancak en temiz yüzey kalitesinin elde edilebileceği strateji seçimi, tecrübeye dayalı olarak belirlenebilmektedir.



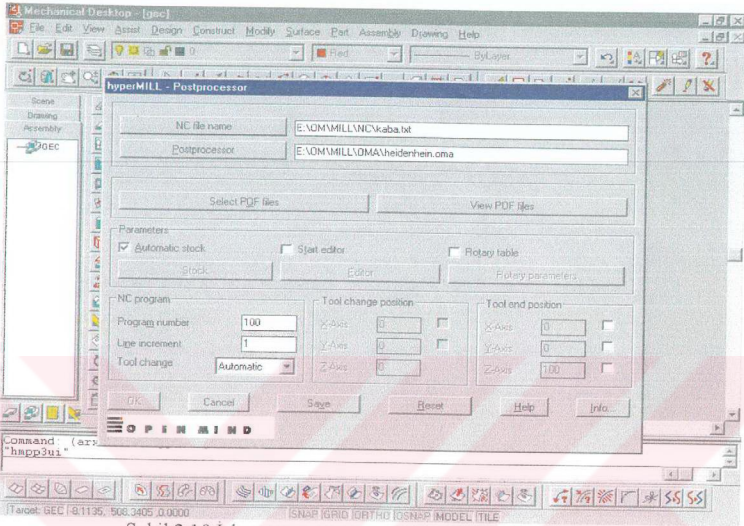
Şekil 2.8 Takım hareket sınırlarının belirtilmesi.

6. CAM programına gerekli bilgilerin verilmesi ardından program takım yollarını çıkaracaktır. Bu işlem sonucunda, takım işleme esnasında hangi hareketleri yapacağı, işleme süresi, hatta takımın yaptığı hareketlerin listesi kullanıcıya verilmektedir (bkz. Şekil 2.10). Kullanıcı isterse işlemi simüle ettirebilir ve takım yolunu kabul edip etmeme kararını buna göre verebilir. Şayet kullanıcı takım yollarını uygun bulmazsa geri dönerek strateji değiştirebilir ve en uygun yolu belirler. Seçim işlemi için genel kriterler işleme zamanı, takımın boşa çalışma süresi, takımın inme kalkma sayısı ya da tecrübeye dayalı olarak işlem stratejisinin yüzey kalitesine olan etkisidir.

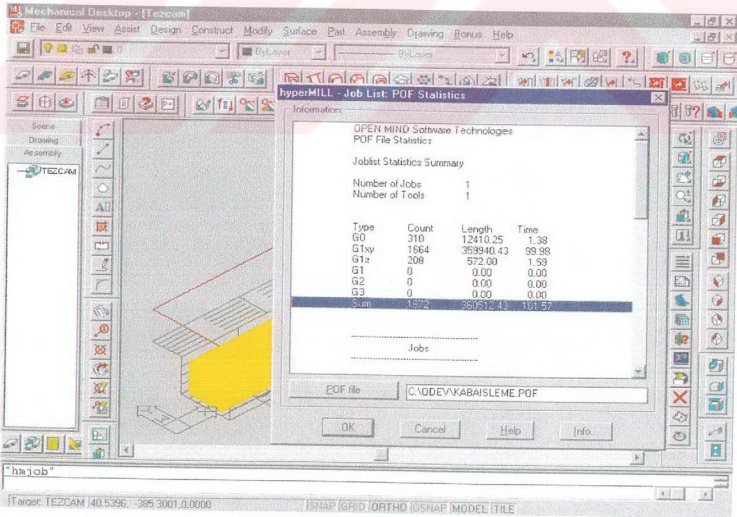


Şekil 2.9 İşleme stratejisinin belirtilmesi.

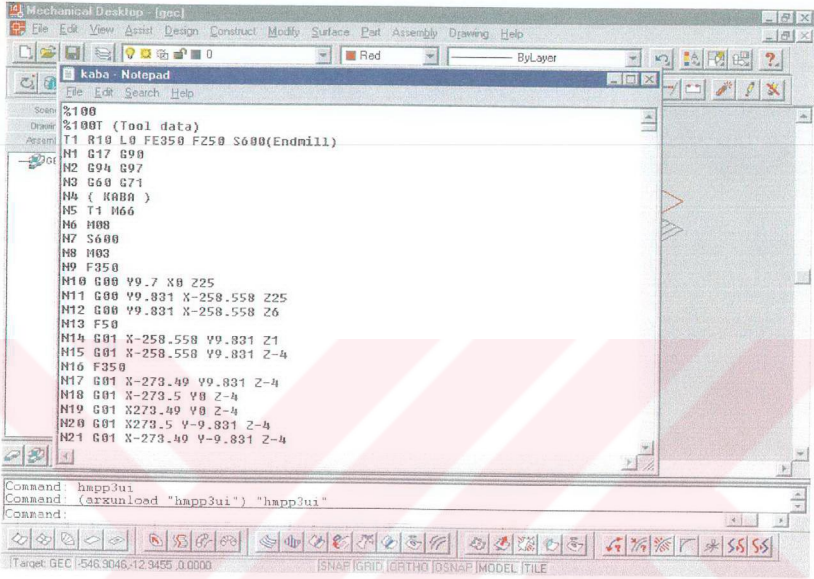
7. Program tarafından çıkarılan takım yollarının kullanıcı tarafından kabul edilmesi ardından, program bu takım yollarının kodlarını, CNC işleme merkezi işlemcisinin tanıyacağı bir formata dönüştürmelidir. Bu amaçla programın post-prosesörü kullanılır (bkz. Şekil 2.11). Genelde çoğu CNC işlemcisi Din-Is0 standart post-prosesörünü tanımaktadır ve bu post-prosesör CAM işlemleri ile beraber gelmektedir. Şayet CNC işlemcisi Din-Is0 kodlarını tanıymıyorsa, özel post-prosesör yazılmalıdır. Post-prosesörün temini genelde, CAM programının alındığı program bayiiilerinden yapılmaktadır. Her CNC işlemcisinin (Heidenhein, Fancu, Siemens gibi) kendine ait bir kod sistemi mevcut olduğundan, bu dönüşümün yapılması gerekli olmaktadır. Dönüştürülen kodlar .txt ya da .nc uzantılı bir metin dosyasında saklanırlar (bkz Şekil 2.12) ve CNC işlemcisine direkt hat vasıtası ile ya da disket ile nakledilirler.



Şekil 2.10 İşleme sonucu program tarafından verilen rapor.



Şekil 2.11 Elde edilen takım yollarını CNC kodlarına dönüştürülmesi



Şekil 2.12 Post-Prosesör tarafından dönüştürülen CNC kodları.

CAD/CAM programları yıllarca iş istasyonlarında UNIX işletim sisteminde çalışabilen programlar olarak yazıldılar. Bu durum, CAD/CAM yatırım maliyetlerini oldukça artırmaktadır. Ancak son yıllarda kişisel bilgisayarların büyük atığa geçerek çok güçlü bilgisayarları daha ucuz fiyatlarla pazara sürmeleri, birçok yazılım tipinde olduğu gibi CAD/CAM yazılımlarının da WINDOWS ortamına taşınmalarını sağladı. Günümüzde CAD/CAM programlarının modelleme ve işleme özelliklerinin dışında olarak seçim kriteri sayılabilecek özelliklerden biri de WINDOWS tabanlı ya da UNIX tabanlı olmasıdır. WINDOWS tabanlı olan CAD/CAM yazılımları, düşük maliyetli ve çok amaçlı kullanılması mümkün olan donanımlarda çalışabildikleri için genelde tercih edilirler.

2.3 Bilgisayar Destekli Analiz (CAE)

Bilgisayarların ucuzlaması ve yaygınlaşmasına paralel olarak bilgisayar ortamında bir çok mühendislik çalışması yapılır hale gelmiştir. Bilgisayar ortamında hareketli mekanizmaların simülasyonu, enjeksiyon kalıplarında enjekte edilen parçanın akış analizleri gibi uygulamalar bunlara örnek olarak verilebilirler. Ancak bilgisayar ortamında mühendislik çalışmaları denince akla ilk olarak Sonlu Elemanlar Analizi gelmektedir. Bir parça, ya da kalıbın Sonlu Elemanlar Analizi ile öngörülen şartlarda mekanik davranışlarının tesbiti işlemi, bilgisayar yardımı olmadan uygulanması çok zordur. Son yıllarda bilgisayarların ve bilgisayar programcılığının çok gelişmesi sonucu çok basit kullanımlı ve hızlı sonuç verebilen bilgisayar programları ortaya çıkmıştır. Bunlara örnek olarak ANSYS, MARC, NASTRAN VE ABAQUS verilebilir. Bu programlar sayesinde pratik uygulamada uygulamacılar Sonlu Elemanlar Analizinin teorisine çok hakim olmak sorunda değildirlir. Zira bu programlar çok basit bir CAD veya CAM programı görünümündedirler. Ancak Sonlu Elemanlar Analizi teorisi hakkında hiç ya da çok az bilgi sahibi bir uygulamacının bu programlarla çok yanlış sonuçlar elde etmesi ve yanılması mümkündür. Dolayısıyla söz konusu paket programlarla doğru sonuç verecek analizlerin yapılabilmesi için iyi bir mühendislik vizyonu (özellikle sistem modelinin çıkarılmasında çok önemli), iyi bir mekanik bilgisi (daha ince ağ örülerek daha hassas çözüm elde için kritik bölgelerin tayin edilmesi amacı ile), ve doğru eleman tiplerinin seçilebilmesi amacı ile yeteri kadar sonlu elemanlar teorisi bilgisi gerekmektedir. Bu nedenle bu bölümde Sonlu Elemanlar Analizi hakkında teorik bilgiler verilecek ve pratik uygulama örneği gösterilecektir.

2.3.1 Sonlu elemanlar analizi teorisi

Sonlu Elemanlar Analizi, analitik metodlarla çözülmesi mümkün olmayan çeşitli problemlerin matematiksel ifadelerle modellenmesi ve bunların yaklaşık olarak çözümlenmesi için kullanılan en yaygın methodtur. Bu yöntem, mikro bilgisayarların gelişmesi ve yaygın olarak kullanılmaya başlanması ile çok hızlı bir şekilde gelişmiş ve birçok alanda uygulama alanı bulmuştur.

Genel olarak Sonlu Elemanlar Analizi Yönteminde, problemin çözüm bölgesi, sonlu elemanlar adı verilen çözüm bölgelerine ayrılır. Bilinmeyenler her bir elemanın genellikle düğüm noktalarında aldıkları değerler cinsinden ifade edilerek eleman denklemleri elde edilir. Eleman denklemlerinin birleştirilmesi ile sistem denklemleri ve genellikle simültane cebrik denklemler formunda bir denklem sistemi elde edilir. Bu denklemlerin sınır koşulları ile birlikte çözümü ile

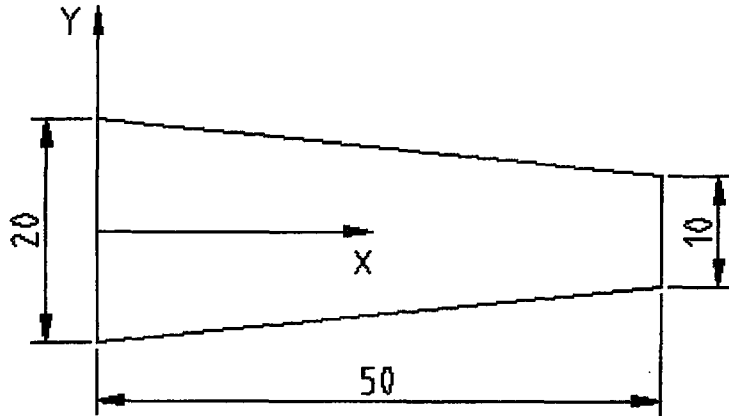
bölge içinde bilinmeyenler yaklaşık olarak elde edilebilir. Bir problemin sonlu elemanlar analizi ile çözümünde aşağıdaki temel adımlar uygulanır;

1. Sürekli sistemin alt bölgelere (elemanlara) bölünmesi.
2. Elemanların karakteristik matrislerinin tanımlanması.
3. Elemanlar için tanımlanan karakteristik matrislerin birleştirilerek sistem karakteristik matrisinin elde edilmesi.
4. Sisteme sınır koşullarının uygulanması.
5. Sistemin çözümü.

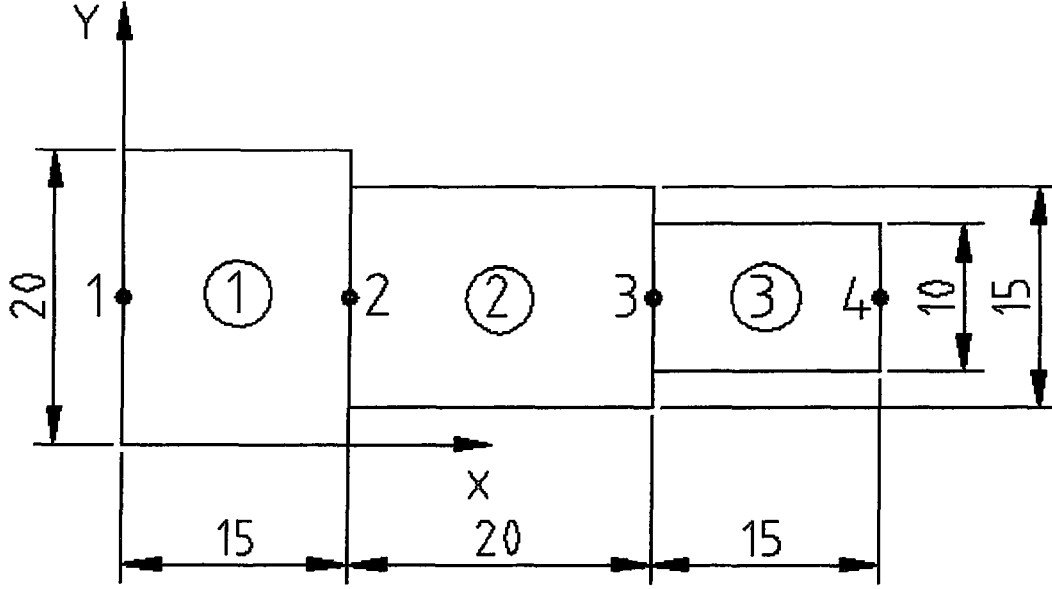
Sonlu elemanlar analizi kendi içinde de çeşitli metodlara ayrılmaktadır. Bu metodlar eleman katılık matrislerinin eldesinde birbirinden farklılıklar göstermektedir. Diğer aşamalar aynıdır. Sonlu Elemanlar Analizi kendi içinde aşağıdaki metodlara ayrılmaktadır;

2.3.1.1 Direkt metod

Direkt Metod, fiziksel tepki temeline dayanmaktadır (Cook, 1988). Çok basit elemanların kullanımı ile sınırlıdır ama sonlu elemanlar analizi mantığının anlaşılması için en kullanışlı metodtur. Bu metodun açıklaması, basit bir çubuk probleminin çözümü ile gösterilebilir. Şekil 2.13'de gösterilen konik çubuğun direkt metod ile çözümünde yapılması gereken ilk işlem bu çubuğun daha basit elemanlardan oluşan alt bölgelere ayrılmasıdır (bkz. Şekil 2.14). Bunun ardından her bir alt eleman için karakteristik matrisleri elde edilir (bkz eşitlik 2.1, 2.2 ve 2.3).



Şekil 2.13 Konik çubuk



Şekil 2.14 Konik çubuğun basitleştirilmiş modeli.

1 nolu eleman için;

$$k_1 \cdot x_1 - k_2 \cdot x_2 = f_1$$

$$-k_1 \cdot x_1 + k_2 \cdot x_2 = f_2$$

$$k_1 = A_1 \cdot E / L_1$$

matris formunda yazacak olursak;

$$k_1 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2.1)$$

aynı şekilde 2 ve 3 nolu elemanlar için;

$$k_2 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2.2)$$

$$k_3 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2.3)$$

elde edilir. Böylece elimizdeki üç elemanın katılık matrisleri elde edilmiştir. Bu katılık matrisleri birleştirilerek genel katılık matrisi elde edilir (eşitlik 2.4);

$$\begin{bmatrix} k_1 & -k_1 & 0 & 0 \\ -k_1 & k_1 + k_2 & -k_2 & 0 \\ 0 & -k_2 & k_2 + k_3 & -k_3 \\ 0 & 0 & -k_3 & k_3 \end{bmatrix} \quad (2.4)$$

Sistemin katılık matrisinin eldesinden sonra sıra, sınır koşullarının ve yüklerin uygulanarak sistemin çözümüne gelmiştir (eşitlik 2.5).

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{Bmatrix} = \begin{bmatrix} k_1 & -k_1 & 0 & 0 \\ -k_1 & k_1 + k_2 & -k_2 & 0 \\ 0 & -k_2 & k_2 + k_3 & -k_3 \\ 0 & 0 & -k_3 & k_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} \quad (2.5)$$

Yukarıda matrisel formdaki eşitliklere sınır koşulları ve başlangıç koşulları uygulanırsa dört bilinmeyenli dört denklem elde etmiş oluruz ve bunun çözümü ile bilinmeyenler elde edilir.

Direkt Metod, daha önce de bahsedildiği gibi çok basit elemanlar kullandığı için pek tercih edilen bir metod değildir. Hiç bir bilgisayar programı bu metodu kullanarak çözüm yapmamaktadır.

2.3.1.2 Varyasyonel formülasyon

Çerçeve sistemleri gibi birbirinden ayrık bileşenlerden oluşan sistemler, sistemi oluşturan bileşen sayısına bağlı olarak kesin sayıda serbestlik derecesi ile modellenebilirler ve bu serbestlik dereceleri de bileşenlerin bağlantı noktalarındaki deplasmanlardır. Elastik bir çubuk gibi sürekli sistemler ise, bünyelerinin her noktasında deplasman olacağından sonsuz sayıda serbestlik derecesine sahiptir. Bu nedenle sürekli sistemlerin davranışları kısmi diferansiyel denklemlerle

ifade edilir. Basit sürekli sistemler için, bu kısmi diferansiyel denklemleri çözen ve sınır koşullarını sağlayan gerilme alanları ya da yerdeğiştirme alanları bulunabilir. Ancak bu tip bir problemi tanımlayabilen bir fonksiyonele (Π_P) Rayleigh-Ritz metodu uygulanarak, kısmi diferansiyel denklemler çözmeye gerek kalmadan, yeterli hassasiyetle yaklaşım yapma yoluna gidilebilir. Bu tip bir yaklaşım yapıldığında problemimiz sonsuz sayıda serbestlik derecesine sahip kısmi diferansiyel denklem tipinden, belirli sayıda serbestlik derecesine sahip cebrik eşitlikler haline getirilmiş olur. Bir Rayleigh-Ritz çözümü, nadir olarak kesin sonuç verir ancak çözüm esnasında serbestlik derecesinin artırılması ile yaklaşım hassasiyeti de artırılabilir.

Rayleigh-Ritz Metodu, 1870 yılında Lord Rayleigh'in titreşim problemleri çalışmalarında ortaya çıkmıştır (Cook, 1988). Bu problemde, tek serbestlik derecesine sahip bir yaklaşım alanı kullanılmıştı. 1909'da Ritz, herbiri gerekli sınır koşullarını (örneğin kinematik) sağlayan ve herbiri birbirinden farklı serbestlik derecesi ile ilişkilendirilmiş birkaç fonksiyondan oluşan bir yaklaşım alanı kurma metodunu genelleştirmiştir. Ritz bu metodu bir denge problemine (static) uygulamıştır. Bir elastik çubukta yüklerin uygulanması sonucu meydana gelen gerilmeler ve yerdeğiştirmelerin elde edilmesine çalışıldığını düşünelim. Bir noktanın yerdeğiştirmesi u , v ve w yerdeğiştirme bileşenleri ile temsil edilsin. Böyle bir problem için Rayleigh-Ritz çözümü, u , v ve w bileşenleri için yaklaşım alanlarının kurulması ile başlar. Herbir yaklaşım alanı, koordinatların fonksiyonu $f_i = f_i(x,y,z)$ ile değeri daha sonra belirlenecek olan bir a_i katsayısı ile çarpılması şeklinde elde edilir. Burada a_i değerleri "genelleştirilmiş koordinatlar" olarak adlandırılabilir.

$$u = \sum_{i=1}^l a_i f_i \quad v = \sum_{i=l+1}^m a_i f_i \quad w = \sum_{i=m+1}^n a_i f_i \quad (2.6)$$

(2.6) eşitliklerinde yer alan f_i fonksiyonları uygunluk denklemlerini ve sınır koşullarını sağlamalıdır. F_i fonksiyonları genellikle polinomlardır ancak bu fonksiyonlar polinom olmak zorunda değildir. (2.6) nolu eşitliklerden elde edilen yaklaşım alanları, problemi tanımlayan fonksiyoneldeki değişkenlerin yerlerine yazıldığında, fonksiyonel içindeki bilinmeyen değerler sadece a_i değerlerinden ibaret olacaktır. Bu noktada dengede olan bir sistemin potansiyel enerji prensibinden faydalanılacaktır (2.7);

$$\frac{\partial \Pi_P}{\partial a_i} = 0 \quad a_i = 1, 2, \dots, n \quad (2.7)$$

ifade edilir. Basit sürekli sistemler için, bu kısmi diferansiyel denklemleri çözen ve sınır koşullarını sağlayan gerilme alanları ya da yerdeğiştirme alanları bulunabilir. Ancak bu tip bir problemi tanımlayabilen bir fonksiyonele (Π_P) Rayleigh-Ritz metodu uygulanarak, kısmi diferansiyel denklemler çözmeye gerek kalmadan, yeterli hassasiyetle yaklaşım yapma yoluna gidilebilir. Bu tip bir yaklaşım yapıldığında problemimiz sonsuz sayıda serbestlik derecesine sahip kısmi diferansiyel denklem tipinden, belirli sayıda serbestlik derecesine sahip cebrik eşitlikler haline getirilmiş olur. Bir Rayleigh-Ritz çözümü, nadir olarak kesin sonuç verir ancak çözüm esnasında serbestlik derecesinin artırılması ile yaklaşım hassasiyeti de artırılabilir.

Rayleigh-Ritz Metodu, 1870 yılında Lord Rayleigh'in titreşim problemleri çalışmalarında ortaya çıkmıştır (Cook, 1988). Bu problemde, tek serbestlik derecesine sahip bir yaklaşım alanı kullanılmıştır. 1909'da Ritz, herbiri gerekli sınır koşullarını (örneğin kinematik) sağlayan ve herbiri birbirinden farklı serbestlik derecesi ile ilişkilendirilmiş birkaç fonksiyondan oluşan bir yaklaşım alanı kurma metodunu geliştirmiştir. Ritz bu metodu bir denge problemine (static) uygulamıştır. Bir elastik çubukta yüklerin uygulanması sonucu meydana gelen gerilmeler ve yerdeğiştirmelerin elde edilmesine çalışıldığını düşünelim. Bir noktanın yerdeğiştirmesi u , v ve w yerdeğiştirme bileşenleri ile temsil edilsin. Böyle bir problem için Rayleigh-Ritz çözümü, u , v ve w bileşenleri için yaklaşım alanlarının kurulması ile başlar. Herbir yaklaşım alanı, koordinatların fonksiyonu $f_i = f_i(x,y,z)$ ile değeri daha sonra belirlenecek olan bir a_i katsayısı ile çarpılması şeklinde elde edilir. Burada a_i değerleri "genelleştirilmiş koordinatlar" olarak adlandırılabilir.

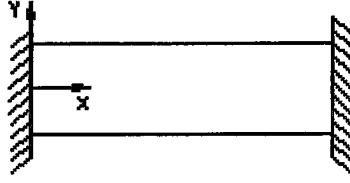
$$u = \sum_{i=1}^l a_i f_i \quad v = \sum_{i=1}^m a_i f_i \quad w = \sum_{i=m+1}^n a_i f_i \quad (2.6)$$

(2.6) eşitliklerinde yer alan f_i fonksiyonları uygunluk denklemlerini ve sınır koşullarını sağlamalıdır. F_i fonksiyonları genellikle polinomlardır ancak bu fonksiyonlar polinom olmak zorunda değildir. (2.6) nolu eşitliklerden elde edilen yaklaşım alanları, problemi tanımlayan fonksiyoneldeki değişkenlerin yerlerine yazıldığında, fonksiyonel içindeki bilinmeyen değerler sadece a_i değerlerinden ibaret olacaktır. Bu noktada dengede olan bir sistemin potansiyel enerjisi prensibinden faydalanılacaktır (2.7);

$$\frac{\partial \Pi_P}{\partial a_i} = 0 \quad a_i = 1, 2, \dots, n \quad (2.7)$$

(2.7) eşitliğindeki işlemler yapıldığında, yaklaşım alanındaki tüm değerler bilinir hale getirilmektedir. Yaklaşım alanlarından sisteme ait yaklaşık yerdeğiştirmeler elde edilebilir. Elde edilen bu yerdeğiştirme değerleri, gerilme-şekil değiştirme bağlantılarında yerlerine konarak sisteme ait gerilme durumuna da yaklaşım yapılabilir.

Bu yaklaşım aşağıdaki örnekle daha iyi anlaşılacaktır (Daloğlu, 1997);



Şekil 2.15 İki uçtan mesnetlenmiş lineer elastik çubuk.

Şekil 2.15' de gösterilen tek boyutlu lineer elastik bir çubuk için potansiyel enerji (fonksiyonel) eşitlik (2.8) da verilmektedir;

$$\Pi = \frac{1}{2} \int_0^L EA \left(\frac{du}{dx} \right)^2 dx - 2U_1 \quad (2.8)$$

şeklinde verilmektedir. Bu sistem için bir yerdeğiştirme polinomu seçeceğiz (bkz. eşitlik 2.9).

$$\tilde{U}_{(x)} = C_1 + C_2 x + C_3 x^2 \quad (2.9)$$

sınır koşullarını bu yerdeğiştirme polinomuna (alanına) uygularsak;

$$X=0 \text{ da } U=0 \text{ ise } C_1 = 0$$

$$X=2 \text{ de } U=0 \text{ ise } 0=C_1+2C_2+4C_3 \text{ ise } C_2 = -2C_3$$

Yerdeğiştirme polinomu (alanı) (2.10) eşitliğindeki şekli alır.

$$\tilde{U}_{(x)} = C_3(-2x + x^2) \quad (2.10)$$

Eşitlik (2.10)'da verilen yerdeğiştirme polinomunu fonksiyonelimizde yerine koyarsak;

$$\frac{d\tilde{u}}{dx} = 2C_3(-1 + x) \quad \Pi = \frac{1}{2} \int_0^L f(x, \tilde{u}(x), \tilde{u}'(x)) dx$$

ise fonksiyonelimiz eşitlik (2.11)'de verilen şekli alır;

$$\Pi = \frac{1}{2} \int E.A.4.C_3^2(-1+x)^2 dx - 2(-C_3) \quad (2.11)$$

$x=1$ için $\tilde{u}_{(1)} = -C_3$ elde edilir. $E = A = 1$ kabul edilirse fonksiyonelimiz eşitlik (2.12)'deki şekilde elde edilir;

$$\Pi = 2C_3^2 \int_0^2 (1-2x+x^2) dx + 2C_3 \quad (2.12)$$

integrasyon yapıldıktan sonra sistemin potansiyel enerji fonksiyoneli eşitlik (2.13)'deki şekilde elde edilir;

$$\Pi = 2C_3^2 \left(\frac{2}{3} \right) + 2C_3 \quad (2.13)$$

sistemin dengede olması nedeni ile bu fonksiyonelin, bilinmeyen sabit sayılara göre diferansiyelleri alınarak bilinmeyenler;

$$\frac{\partial \Pi}{\partial C_3} = \frac{8}{3}C_3 + 2 = 0 \Rightarrow C_3 = \frac{-3}{4} \quad (2.14)$$

şeklinde elde edilir. Elde ettiğimiz sabit sayıları yerdeğiştirme fonksiyonunda yerlerinde koyarsak, yerdeğiştirme fonksiyonumuzu eşitlik (2.15)'deki şekilde elde ederiz;

$$\tilde{U}_{(x)} = -\frac{3}{4}(-2x + x^2) \quad (2.15)$$

Bu fonksiyonu kullanarak şekildeki çubuğun uzunluğu boyunca herhangi bir x mesafesindeki yerdeğiştirmesini elde edebiliriz.

Varyasyonel formülasyonun bazı avantajları aşağıdaki gibi özetlenebilir;

- Varyasyonel metotta sistem denge denklemlerinin oluşturulması göreceli olarak daha kolaydır. Bu kolaylığın en büyük nedeni, varyasyonel formülasyonda kullanılan değişkenlerin vektörel değişkenlerden ziyade (kuvvet, yerdeğiştirme vs.) skaler değişkenler olmasıdır (enerji, potansiyel vs.).
- Direkt metotta sisteme etkileyen tüm etkenler formülasyona dahil edilmelidirler. Ancak varyasyonel formülasyonda böyle bir zorunluluk yoktur. Sistemde net iş yapmayan etkenler, varyasyonel formülasyonda göz önüne alınmazlar.

2.3.1.3 Ağırlıklı kalıntılar yöntemi

Varyasyonel metotta fonksiyonel adı verilen bir integrasyon denklemi kullanılmaktadır. Ancak bazı durumlarda böyle bir fonksiyonele sahip elde etmek mümkün olmamaktadır. Böyle bir durumda Ağırlıklı kalıntılar metodunu kullanarak yaklaşım yapılır (Bathe, 1982). Bu metod Sonlu Elemanlar Analizinin temelini oluşturmaktadır. Daha önce bahsedilen metodlarda çözümler gerçek çözümlerdir ancak ağırlıklı kalıntılar yaklaşımında sonuçlar yaklaşık sonuçlardır ve sonuçların hassasiyeti analist tarafından belirlenir.

Bu metodu açıklamak için statik bir problemi ele alalım.

$$L_{2m}[\phi] = r \quad (2.16)$$

Burada L_{2m} lineer diferansiyel operatörü, ϕ durum değişkeni ve r kuvvet fonksiyonunu belirtmektedir. Problemin çözümü sınır koşullarını da sağlamalıdır;

$$B_i[\phi] = q_i \Big|_{\text{sınır}} \quad i=1, 2, \dots \quad (2.17)$$

Bu methoda simetriklik, çözüm stratejisinde çok fazla önem kazanmaktadır. Bu nedenle burada simetrik ve pozitif tanımlı operatör kavramını açıklamak yerinde olacaktır. Eğer bir lineer diferansiyel operatörü simetrik ise aşağıdaki durum geçerli olacaktır;

$$\int_D (L_{2m}[u])v dD = \int_D (L_{2m}[v])u dD \quad (2.18)$$

Eğer lineer operatör pozitif tanımlıysa aşağıdaki durum da geçerli olacaktır;

$$\int_D (L_{2m}[u])u dD > 0 \quad (2.19)$$

Burada D, operatörün geçerli olduğu bölgeyi, u ve v de gerek ve yeter sınır şartlarını sağlayan fonksiyonlardır.

Ağırlıklı kalıntılar metodunda temel adım aşağıdaki formda bir çözümün kabulüdür;

$$\bar{\phi} = \sum_{i=1}^n a_i f_i \quad (2.20)$$

Burada f_i lineer bağımsız deneme fonksiyonlarını ve a_i de çözüm aşamasında belirlenecek çarpanları temsil etmektedir. Ağırlıklı kalıntılar metodu direkt olarak (2.16) ve (2.17) eşitliklerine etki eder. Bu metodu kullanarak (2.20) deki f_i 'leri, (2.17)'deki sınır koşullarını sağlayacak şekilde seçeriz. Bunun ardından aşağıda gösterilen şekilde kalıntıları hesaplarız;

$$R = r - L_{2m} \left[\sum_{i=1}^n a_i f_i \right] \quad (2.21)$$

Çözümün kesin sonuç verebilmesi için kalıntıların sıfır olması gerekmektedir. İyi bir çözüm için R kalıntıları, problemin çözüm alanı içerisinde yeterince küçük olmalıdır. Ancak ağırlıklı kalıntılar metodunun tüm çözüm şekillerinde a_i 'ler, R değeri yok olacak şekilde seçilirler.

Ağırlıklı kalıntılar metodunda en önemli adımlardan birisi de a_i 'lerin eldesi için eşzamanlı eşitliklerin çözülmesi işlemidir. Bu amaçla kullanılan metodlardan en popüler olanları aşağıda verilmiştir;

- **Galerkin Metodu:** Bu methoda a_i 'ler n sayıdaki eşitlikten elde edilirler;

$$\int_D f_i R dD = 0 \quad i = 1, 2, \dots, n. \quad (2.22)$$

Burada D çözüm bölgesidir.

- **En Küçük Kareler Metodu:** Bu methoda kalıntının karesinin integrasyonu a'ya göre diferansiyelleri alınarak minimize edilir.

$$\frac{\partial}{\partial a_i} \int_D R^2 dD = 0 \quad i = 1, 2, \dots, n \quad (2.23)$$

L_{2m} lineer bir operatör olduğu müddetçe yukarıdaki metodların hepsinde de a_i parametrelerinden oluşan lineer eşitlikler elde edilebilir. Eğer L_{2m} simetrik ve pozitif tanımlı ise Galerkin metodundaki katsayılar matrisi de simetrik ve pozitif tanımlıdır. En küçük kareler metodunda ise L_{2m} operatörünün özelliklerinden bağımsız olarak her zaman simetrik katsayılar matrisi elde edilir.

Yukarıda açıklanan metodları kullanarak direkt olarak (2.16) ve (2.17) eşitliklerine etki ederiz ve böylece (2.20)'deki deneme çözümü ile gerçek çözüm arasındaki hata minimize edilir.

2.3.1.4 Gerilmelerin hesabı

Yukarıda bahsedilen metodlardan herhangi biri kullanılarak yapılan sonlu elemanlar çözümü sonucunda, sisteme ait herbir düğüm noktasındaki yerdeğişmeler bilinmektedir. Sistemin gerilme durumunun incelenmesi için bu yerdeğişirme değerlerinden faydalanılır. Gerilme hesabı aşağıdaki eşitliğin çözümü ile elde edilebilir (Bathe, 1982);

$$\{\sigma\} = [E](\{\epsilon\} - \{\epsilon_0\}) + \{\sigma_0\} \quad (2.24)$$

Burada ϵ_0 ve σ_0 sırasıyla başlangıç durumundaki şekil değişimini ve ön gerilmeyi temsil etmektedir.

2.3.2 Bilgisayar destekli sonlu elemanlar analizi

Bir önceki bölümde Sonlu Elemanlar Analizinin teorisinden bahsedilmişti. Görüldüğü üzere basit sistemler için bu tip hesaplamaların yapılması mümkün olurken karmaşık problemlerde bilgisayar yardımı olmaksızın bu hesaplamaların yapılması mümkün olmamaktadır. Bu nedenle endüstri uygulamalarında sonlu elemanlar analizi yardımı ile hesaplamaların yapılabilmesi için bu amaca hizmet eden çeşitli programlar mevcuttur. Bu programların birbirlerine göre çeşitli avantaj ve dezavantajları vardır. Ancak bir programın popüler olabilmesi için temel kriterler kullanıcı arayüzünün kullanışlı olması, program menülerinin basit ve anlaşılır olması, hızlı çözüm yapabilmesi, barındırdığı eleman sayısı ve sonuçların anlaşılır ve görsel olarak sunum yapmaya elverişli olmasıdır. Genelde bu programların her birinin kendi preprosesör, çözücü ve postprosesörleri vardır (MARC, ANSYS, NASTRAN gibi). Ancak ABAQUS gibi bazı programlar genelde sadece çözümleyici olarak kullanılırlar. Bu programlar için modelin hazırlanması diğer programların preprosesörlerinde yapılır ancak çözüm ABAQUS'a yaptırılır. Çözüm sonuçları yine başka bir postprosesörde çizdirilir. Hatta bu yöntem o kadar çok yaygınlaşmıştır ki sadece pre-prosesör ve postprosesör yazılımları vardır. Bir diğer analiz program çeşitleri de bazı CAD sistemlerine entegre olarak çalışan programlardır (DESIGN SPACE, NASTRAN gibi). Bu programlar, tasarım amacı ile kullanılan CAD programının altında bir menü olarak yer alır ve CAD programının ortamını kullanırlar. Kendilerine ait bir arayüzleri yoktur ancak çok ciddi amaçlı hesaplamalar için pek kullanılamazlar çünkü bu tip programlar genelde sadece statik analizi içerirler ve eleman çeşitleri pek fazla değildir. Yukarıda bahsedilen programların menüleri ve kullanım prosedürleri çok farklı olabilmektedir ancak hepsinin de bir

problemi çözüm aşaması genel olarak aynıdır. Bu aşamalar aşağıdaki gibi sıralanabilir (Cook, 1989);

1. Sürekli alanın ya da parçalı sistemin sonlu elemanlara bölünmesi. Bu işlem programların pre-prosesörleri tarafından yapılır.
2. Her bir elemanın özelliklerinin formüle edilmesi. Bunun anlamı, seçilen eleman tipinde izin verilen deformasyona uygulanabilecek yük tiplerinin belirlenmesidir.
3. Sistemin sonlu eleman modelinin elde edilebilmesi için bu elemanların birleştirilmesi
4. Bilinen noktasal yüklerin, momentlerin, yüzey basınçlarının uygulanması.
5. Bu aşamada yapının nasıl mesnetlendiği belirtilir. Burada bazı düğüm noktalarına, değeri genelde sıfır olan yerdeğiştirme değerleri verilir.
6. Sistemin yerdeğiştirmelerini bulmak için lineer cebrik eşitliklerin çözülmesi işlemi yapılır.
7. Hesap sonucu elde edilen yerdeğiştirme değerleri kullanılarak, incelenen sistemin şekil değişimi ve gerilme durumu hesaplanır. Programın postprosesör adı verilen kısmı ile sonuçların kullanıcıya aktarılması işlemi yerine getirilir.

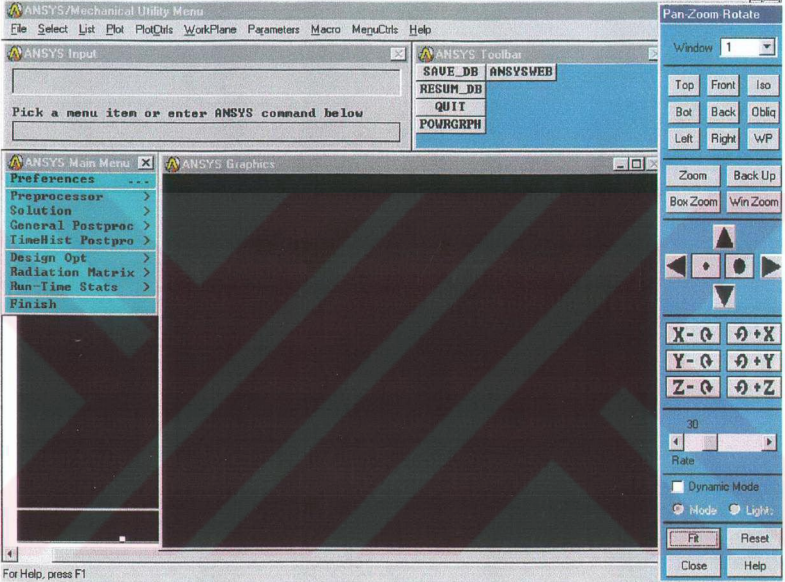
Yukarıdaki aşamalarda program, sadece 1., 4. ve 5. aşamalarda kullanıcıya ihtiyaç duymaktadır. Diğer aşamaların tümü sadece program tarafından gerçekleştirilir. Bu bölüm içerisinde yapılacak olan hesaplamalarda bu aşamalar net olarak görülecektir.

Bölüm 1.1'de tasarımı yapılmış olan delme kalıbında en kritik elemanlar şüphesiz dışi mührerlerdir. Bu bölümde ANSYS Sonlu Elemanlar Analizi programı kullanılarak bu kritik elemanların gerilme durumları incelenecek ve sonuçlar kısmında bu analiz sonuçları irdelenecektir.

2.3.2.1 Kullanılacak programın (ANSYS) tanıtılması

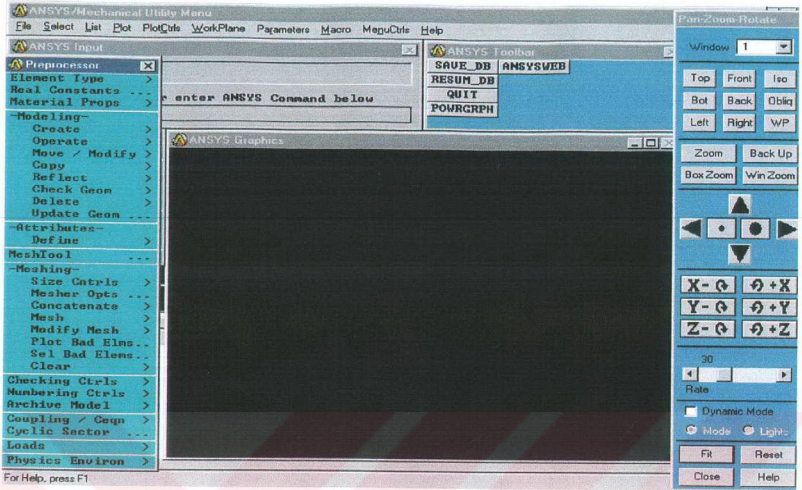
ANSYS Sonlu Elemanlar Analiz Programı, 1970 yılında Dr. John A. Swanson tarafından kurulan Swanson Analysis Systems şirketi tarafından piyasaya sürülen bir sonlu elemanlar paket programıdır (SAS, Inc., 1996). Şu anda dünyada ki en popüler sonlu elemanlar analizi programları arasında yerini almaktadır. Program pre-prosesör, prosesör (solver) ve post-prosesör modüllerinin üçünü de içermektedir. Kolay kullanım özellikleri ve zengin eleman kütüphanesi ile özellikle uçak, otomotiv, tıbbi cihazlar, elektronik cihazlar, endüstriyel ürünler sanayiilerinde yaygın olarak kullanılmaktadır. Ansys ile mekanik analiz, ısı transferi analizi, akışkanlar

mekanik analizi ve elektromanyetik alan analizi yapılması mümkündür. Ayrıca kullanıcıya statik analiz olanakları yanısıra dinamik analiz yapma olanakları, kompozit malzemelerle analiz yapabilme olanakları da ANSYS programı tarafından sunulmaktadır. Burada yapılacak olan analizlerde ANSYS 5.4 Windows NT sürümü çözüm için kullanılacaktır. Şekil 2.16'da ANSYS 5.4'ün genel olarak görünümü verilmektedir.

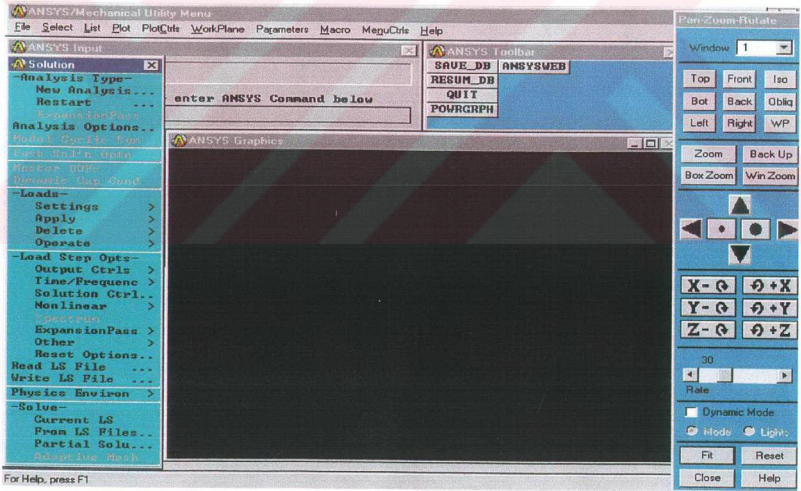


Şekil 2.16 ANSYS 5.4 arayüzünün genel görünümü.

Şekil 2.16'dan görüldüğü gibi sol kısımda bulunan ana menüde preprosesör (preprocessor), çözücü (Solution) ve postprosesör (General Postproc) alt menüleri görülmektedir. Genel olarak bir analizi kullanımında bu üç menüden faydalanılmaktadır. Şekil 2.17'de preprosesör altmenüsünün içeriği görülmektedir. Preprosesör altmenüsü içerisindeki alt menülere bakıldığında eleman tipinin, malzeme özelliklerinin, analizi yapılacak sistem tasarımının ve sistemin alt elemanlara bölünmesi işleminin bu menüden yapılacağı anlaşılabilmektedir.



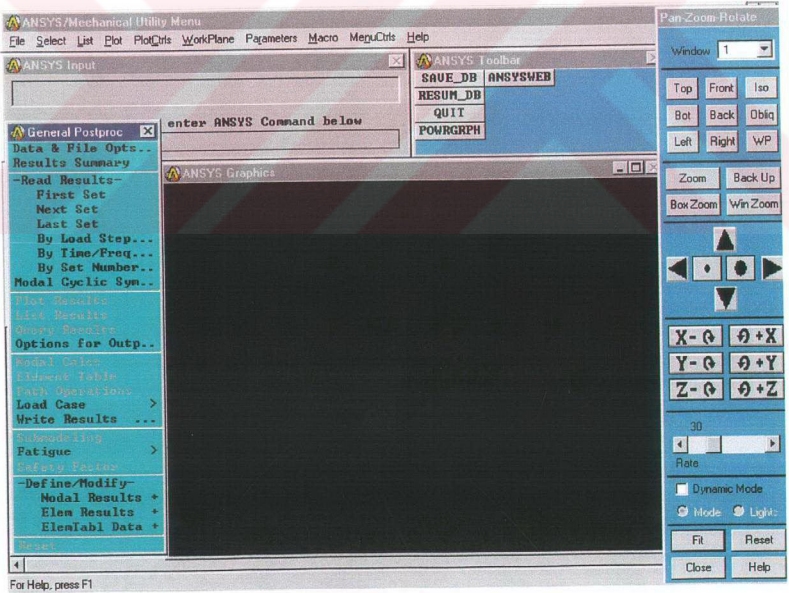
Şekil 2.17 Prosesör menüsünün içerdiği alt menüler.



Şekil 2.18 Çözüm menüsünün içerdiği alt menüler

Burada en önemli işlem, sistemin alt elemanlara (sonlu elemanlara) bölünmesi işlemidir. Elemanların büyüklük ayarlarını rafine etme işlemleri hep bu menüden (meshtool) yapılmaktadır. Şekil 2.18’de ise çözüm menüsü görülmektedir. Preprosesör menüsündeki işlemler tamamlandıktan sonra çözümün yapılabilmesi amacı ile bu menüden faydalanılır.

Bu menüde yapılacak olan ilk işlem, sisteme etkileyen yüklerin belirlenmesi ve uygulanması işlemidir. Ardında ise sıra sistemin mesnetlenme durumunu belirlemektir. Bu işlemler “loads” menüsünden yapılmaktadır. Eğer zamana göre değişen yükler ya da mesnetleme durumları söz konusu ise bu belirlemeler “load step” options menüsünden yapılmalıdır. Programa gerekli tüm bilgiler verildikten sonra bu menüden analiz başlatılır. Analiz sonuçlandırıldığında Şekil 2.19’da görülen postprosesör menüsünden (General postproc) analiz sonuçlarının görüntülenmesi işlemi gerçekleştirilir. Analiz sonuçlarının okutulması işlemi bu menüden yapılır. Okutulma işlemi sonucu görsel presentasyonların yapılması işlemi için “Plot Results” altmenüsü kullanılır.



Şekil 2.19 Postprosesör menüsünün içerdiği alt menüler

2.3.2.2 Mührelerin sonlu elemanlar analizlerinin yapılması

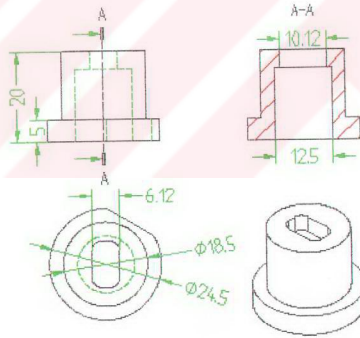
Analizi yapılacak olan kalıbın delegeği malzemenin mekanik özellikleri Çizelge 2.1.'de verilmiştir. Çizelgedeki mekanik özelliklerin değerleri eşitlik (2.25)'deki yerlerine yazılarak gerekli olan mm başına kesme kuvveti: 600 N olarak hesaplanmıştır (Kurt, 1988). Şekil 2.20'de, analizi adım adım gösterilecek olan dişi mührerin geometrisi ve ölçüleri verilmektedir. Mühre malzemesi daha önceden SVERKER 21 olarak verilmiştir.

Çizelge 2.1 Kullanılacak sac malzemesinin mekanik özellikler tablosu (Güneş, 1989).

Sac Malzemesi	Kalınlık (mm)	Çekme Dayanımı (Mpa)	Kesme Dayanımı (Mpa)	Elastisite Modülü (Mpa)
St 42.21	1.5	420-500	400	200000

$$F_k = L.s.\sigma_k$$

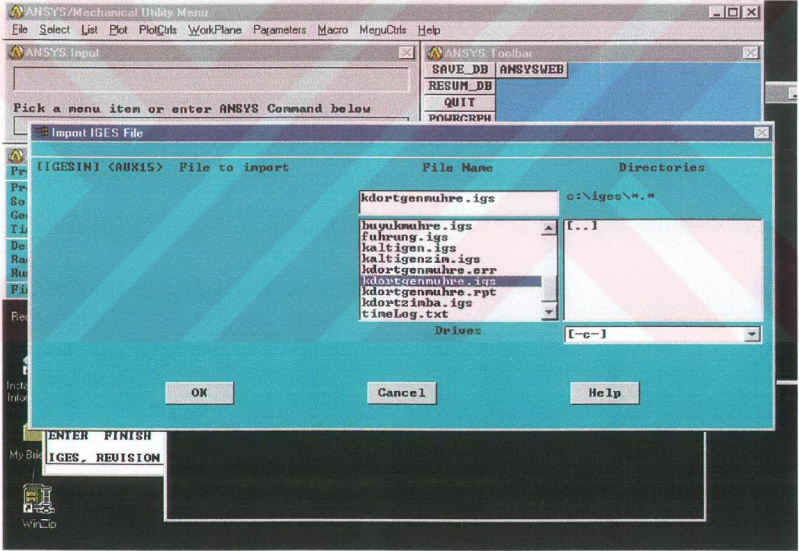
(2.25)



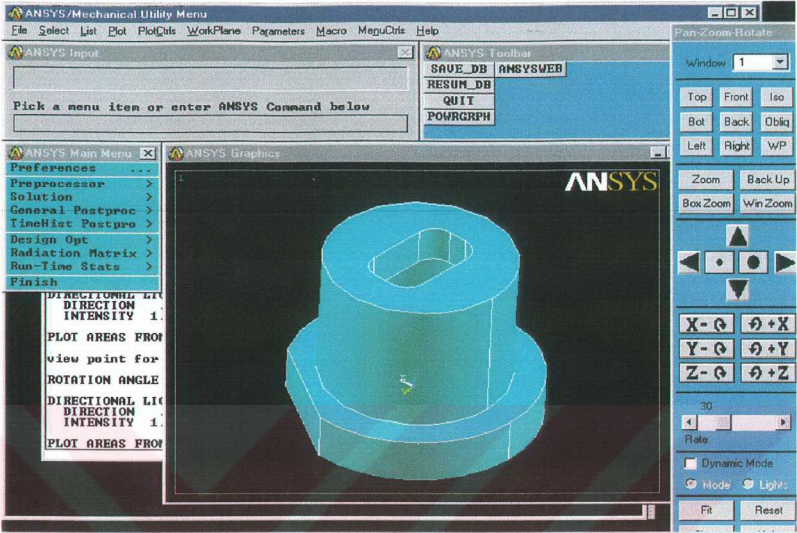
Şekil 2.20 Analizi yapılacak olan mührerin geometrisi ve ölçüleri.

Analizin ilk aşamasında, incelenecek geometrinin ANSYS ortamında oluşturulması işlemi yapılır. Bu işlem iki farklı şekilde gerçekleştirilebilir. İlk seçenek, geometrinin ANSYS ortamında, programın sunduğu çizim araçları kullanılarak analiz tarafından oluşturulmasıdır. Diğer seçenek ise geometrinin başka bir CAD programında katı veya yüzey olarak

oluşturulması ve Ansys ortamına herhangi bir transfer formatı şeklinde (genellikle IGES formatında) transfer edilmesidir. İkinci seçenek daha kolay olduğundan daha çok tercih edilmektedir. Bu analizde de bu seçenek kullanılacaktır. Şekil 2.21'de geometrinin Ansys ortamına transfer edilişi için kullanılan menü adımları görülmektedir. Geometri CAD ortamında katı model olarak oluşturulmuştur. Ancak daha sonra transfer edebilmek amacı ile IGES formatında dönüştürülmüştür. IGES formatı katı model depolamadığından bu katıyı yüzey halinde çevirmiştir. Ansys ise bu yüzeye dönüşmüş modeli okurken kendi ortamında tekrar katıya dönüştürmüştür (bkz Şekil 2.22). Bu işlemler kullanıcının müdahalesine gerek kalmadan CAD programı ve ANSYS tarafından yapılmıştır. Bu dönüşüm esnasında ortaya bazı hataların çıkması durumunda kullanıcı müdahale ederek muhtemel hataları düzeltme olanağına sahiptir.

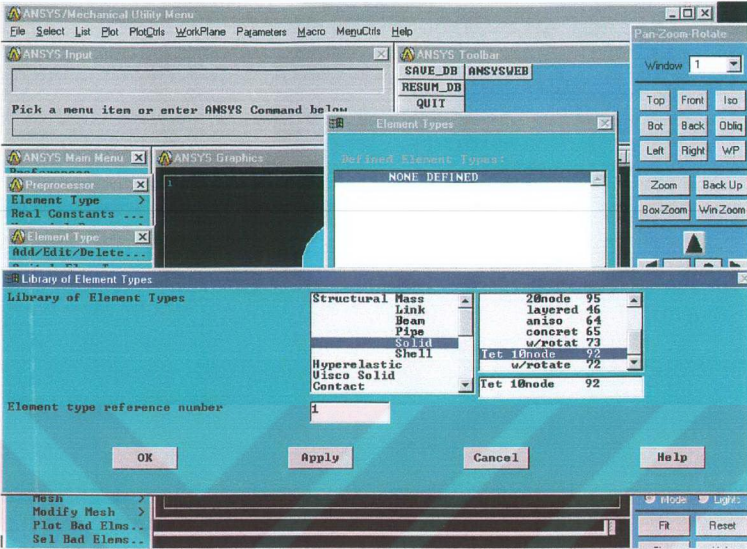


Şekil 2.21 Geometrinin ANSYS ortamına IGES formatı kullanılarak transfer edilmesi.

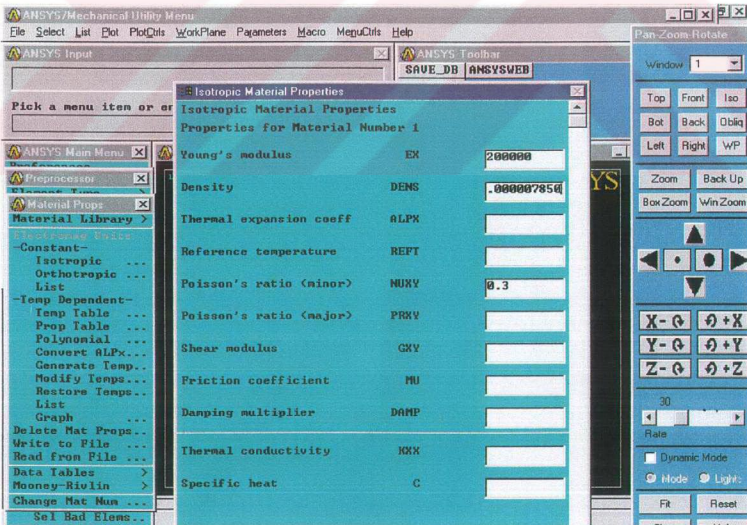


Şekil 2.22 ANSYS tarafından katı modele dönüştürülmüş olan geometri.

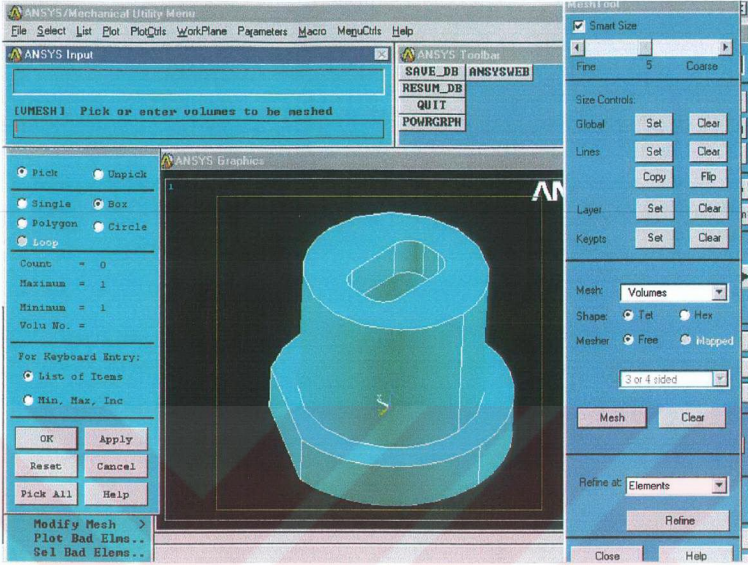
Analizi yapılacak olan geometri ANSYS ortamına taşındıktan sonra, daha önce bahsedilmiş olan analiz aşamaları aynı sırada yapılacaktır. İlk olarak geometrinin sonlu elemanlar bölünmesi işlemi yapılacaktır. Bu aşamada en önemli adım eleman tipinin belirlenmesidir. Modelimiz katı bir model olduğundan “Solid Eleman” tipinde bir eleman kullanmak durumundayız. Şekil 2.23’de de görüldüğü üzere ANSYS bünyesinde birden fazla solid eleman tipi mevcuttur. Her bir elemanın birbirine göre avantaj ve dezavantajları vardır. Burada kullanılacak eleman tipi olarak ise 92 nolu eleman seçilmiştir (Tet 10 Node 92). Bu eleman özellikle başka CAD ortamında transfer edilerek elde edilmiş geometrilerin modellenmesi için tavsiye edilmektedir. Bu eleman hakkında ANSYS yardım menüsünden alınmış daha ayrıntılı bilgiler tez çalışmasının sonunda Ek 1.’de verilmektedir. Eleman tipi seçildikten sonra yapılacak işlem ise bu elemana malzeme özelliklerinin atanması işlemidir. İsootropik bir malzeme kullanılarak mekanik analiz yapılabilmesi için belirlenmesi gereken iki özellik Elastisite Modülü (E) ve Poisson oranıdır (ν) (bkz. Şekil 2.24).



Şekil 2.23 Eleman tipinin seçilmesi aşaması.



Şekil 2.24 Malzeme özelliklerinin belirlenmesi işlemi.

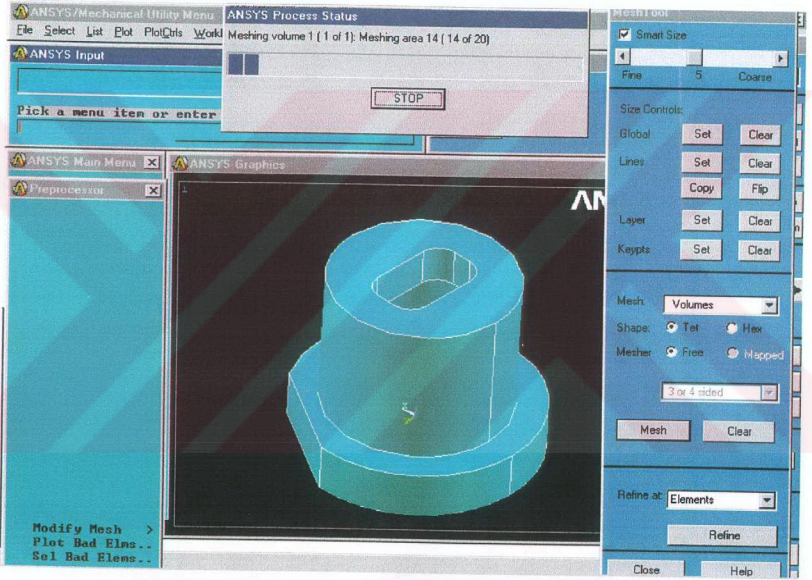


Şekil 2.25 Sonlu elemanlara bölme özelliklerinin belirlenmesi.

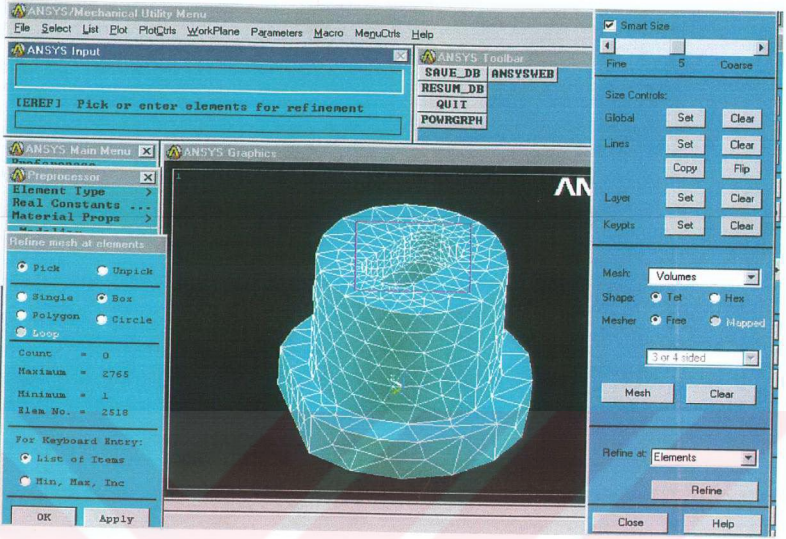
Malzeme özellikleri de belirlendikten sonra sıra geometrinin alt elemanlara bölünmesi işlemine gelmiştir. ANSYS’de alt elemanlara bölme işlemi, “meshtool” adı verilen bir modül yardımı ile yapılır. Bu menüde (bkz Şekil 2.25) eleman büyüklükleri ve hassasiyetleri belirlenir. Gerekli belirlemeler yapıldıktan sonra sonlu elemanlara bölünecek olan geometri seçilir ve alt elemanlara bölme işlemi başlatılır. Bölme esnasında ANSYS kullanıcıya kaç eleman oluşturulacağını ve gerekli ise uyarılarını mesaj penceresinden iletir (bkz. Şekil 2.26) Bir analiz gerçeği olabilmesi için en önemli faktörlerden biri de sonlu elemanlara bölme aşamasıdır. Sonlu eleman boyutlarının büyük olması, analiz süresini kısaltacak ancak elde edilen sonuçların yeterince hassas olmaması sorununu doğurabilir. Eleman boyutlarının küçük seçilmesi durumunda ise sonuçlar hassas olacaktır ancak analiz süresi fazla uzayacaktır. Bu nedenle eleman boyutlarının optimizasyonu çok önemlidir. Ancak genelde analizi yapılacak olan geometrinin belirli kritik bölgeleri için hassas çözüm istenir. Bu bölgeler genelde ani kesit değişimlerinin olduğu bölgeler ya da kuvvetin direkt etki ettiği bölgelerdir. Bu nedenle kritik bölgelerde daha ince bir ağ yapısı (daha küçük elemanlar) istenirken, diğer bölgelerde daha kaba bir ağ yapısına (daha büyük elemanlar) müsaade edilir.

Bu ağ yapısını sağlamak amacıyla geometri ilk olarak kaba elemanlara bölünür ve daha sonra da kritik bölgelerin ağ yapısında iyileştirmeler (refining) yapılır. ANSYS’de bu işlem yine “meshtool” vasıtası ile yapılır (bkz Şekil 2.27).

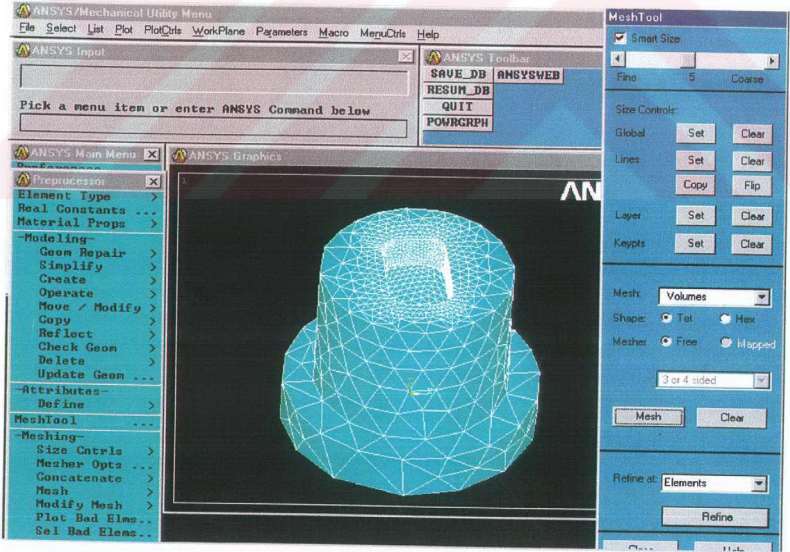
Sonlu elemanlara bölme işlemi ardından “meshtool” daki “refine” butonuna basılarak inceltilecek bölgeler seçilir ve iyileştirme komutu verilir. Seçilen bölgelerdeki elemanlar tekrar bölünerek bölgedeki ağ yapısı inceltilir ve bu bölgelerde daha hassas sonuçların eldesi için gerekli ağ yapısı sağlanmış olur (bkz. Şekil 2.28).



Şekil 2.26 Geometrinin sonlu elemanlara bölünmesi işlemi.



Şekil 2.27 Kritik bölgelerin ağ yapısında iyileştirme işleminin yapılması.



Şekil 2.28 Rafine işlemi yapılmış geometri ve ağ yapısı.

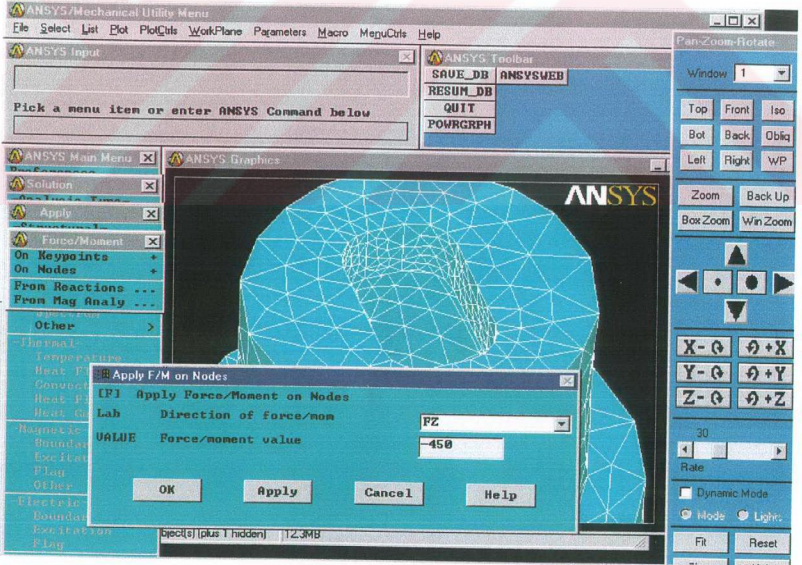
Ağ yapısı oluşturulduktan sonra geometrinin yükleme durumu belirlenmelidir. Söz konusu deliği açmak için mm başına 600N'luk bir kuvvetin gerekli olduğu hesaplanmıştır (bkz. Eşitlik 2.25). Bu değer deliğin kenar uzunluğu ile çarpılarak mühreye etkiyen toplam kuvvet elde edilir. Buna göre mühreye etkiyen kuvvetler,

$$F_{uzunkenar} = 6 \times 600 = 3600 \text{ N}$$

$$F_{kısa kenar} = 2 \times 600 = 1200 \text{ N}$$

$$F_{yuvarlatma} = 3 \times 600 = 1800 \text{ N}$$

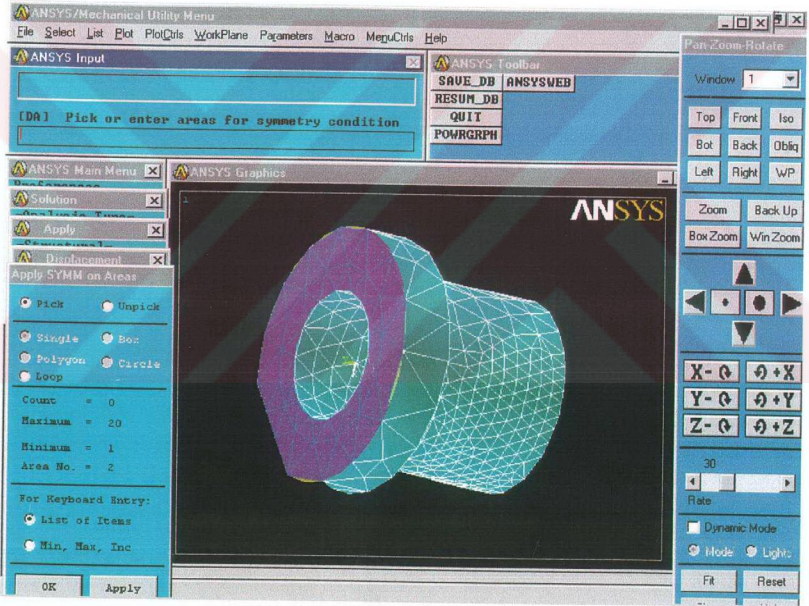
olarak elde edilir. Uzun kenarda 4 eleman oluştuğuna göre burada eleman başına 900 N, kısa kenarda 3 eleman oluştuğuna göre eleman başına 400 N, ve yuvarlatmalarda 3 eleman olduğundan eleman başına 600 N kuvvet düşmektedir. Bu kuvvetler elemanların düğüm noktalarına bölündüğünde uzun kenardaki bir düğüm noktasına 450 N, kısa kenardaki düğüm noktasına 200 N ve yuvarlatmalardaki düğüm noktası 300 N'luk kuvvetler karşılık gelmektedir. Belirlenen bu kuvvetler ilgili düğüm noktalarına noktasal yükler olarak belirtilirler (bkz. Şekil 2.29).



Şekil 2.29 Düğüm noktalarına yüklerin uygulanması.

Yükleme durumunun belirtilmesi için “Solution” menüsünün, “Apply” altmenüsünden “Force/Moment” seçeneği seçilerek farenin ilgili düğüm noktalarına tıklanması vasıtasıyla yüklemeye seçimleri yapılır.

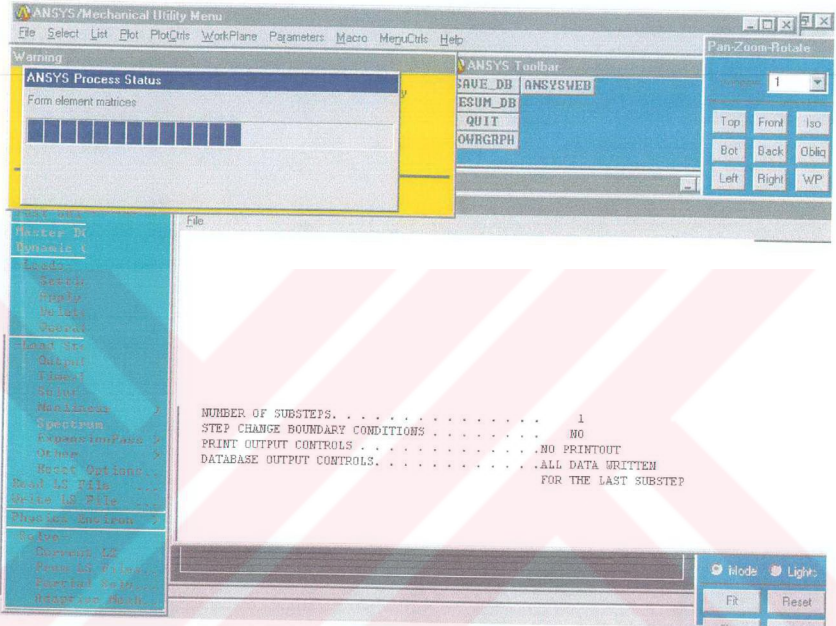
Yüklemeye durumunun belirtilmesi işlemi ardından sıra geometrinin mesnet durumunun belirtilmesine gelmiştir. Mühre, kalıp gövdesi içine sıkı geçme olarak yerleştirildiğinden, üst yüzey hariç tüm yüzeylerin hareketleri sınırlandırılmıştır ve bu nedenle sınırlama, yüzey sınırlaması olarak belirtilecektir. Bu işlemin gerçekleştirilmesi için yine “Solution” menüsünün “Apply” alt menüsünden “Displacement” seçeneği seçilerek sınırlamanın yapılacağı yüzeyler farenin tıklanması vasıtasıyla seçilir (bkz. Şekil 2.30).



Şekil 2.30 Sınırlamaların belirlenmesi işlemi.

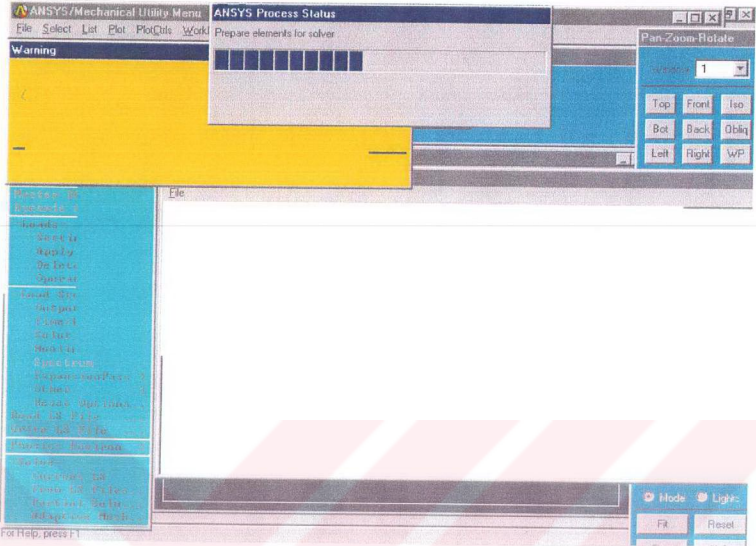
Sınırlama işleminin de bitmesinin ardından analizin yapılabilmesi için gerekli tüm şartlar sağlanmıştır. Bundan sonraki işlemler için program kullanıcı girdisine ihtiyaç duymayacak ve Sonlu Elemanlar Prosedürünü uygulayacaktır. Çözüm işleminin başlatılması için “Solution”

menüsünün “Current LS” seçeneği seçilirse o anda sisteme yüklü olan modelin çözümleme işlemine başlanır. Çözüm işleminin ilk aşamasında herbir eleman için katılık matrisleri oluşturulacaktır (bkz. Şekil 2.31).

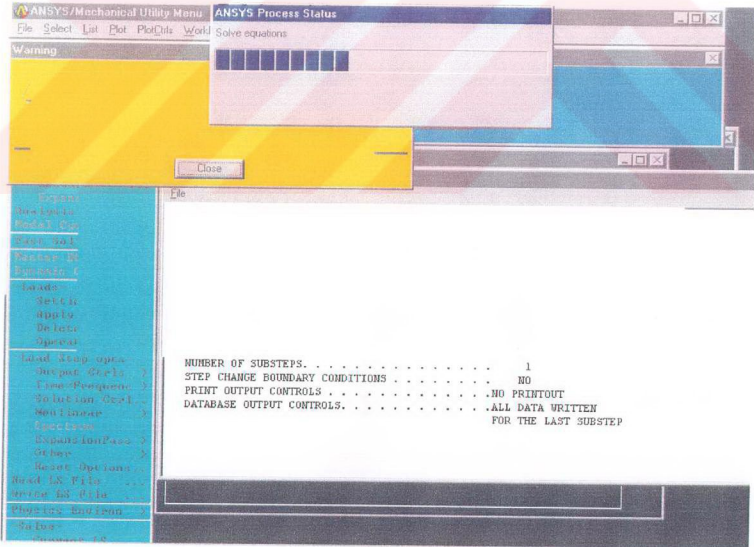


Şekil 2.31 Eleman matrislerinin oluşturulması aşaması.

Herbir eleman için katılık matrislerinin elde edilmesinden sonra program bu katılık matrislerini birleştirecek ve sistemin global katılık matrisini elde edecektir. Elde edilen global matrisle sınır koşulları ve başlangıç yükleri uygulanarak elenebilecek satırlar elenecek ve global matristen çözücünün (solver) hesaplaması için lineer cebrik eşitlik sistemi oluşturulacaktır (bkz. Şekil 2.32). Sonlu elemanlar programlarında çözümleme işleminde lineer cebrik denklem sisteminin çözülmesi amacı ile popüler olarak Gaus Eliminasyon metodu kullanılmaktadır. Gaus Eliminasyon Metodunun bilgisayar program kodları Ek 2. de verilmiştir.



Şekil 2.32 eleman matrislerinin birleştirilerek global matrisin eldesi aşaması.



Şekil 2.33 Lineer cebrik denklem sisteminin çözülmesi aşaması.

seçildiğinde sonuçlar okunacak ve kullanıcının istediği sonuçlar listelenecektir. Biz toplam gerilme durumu ve toplam deformasyon durumu ile ilgilenmekteyiz ve bu nedenle sadece bu iki tip sonucu göstermesini sağlayacağız.



3. SONUÇLAR ve ÖNERİLER

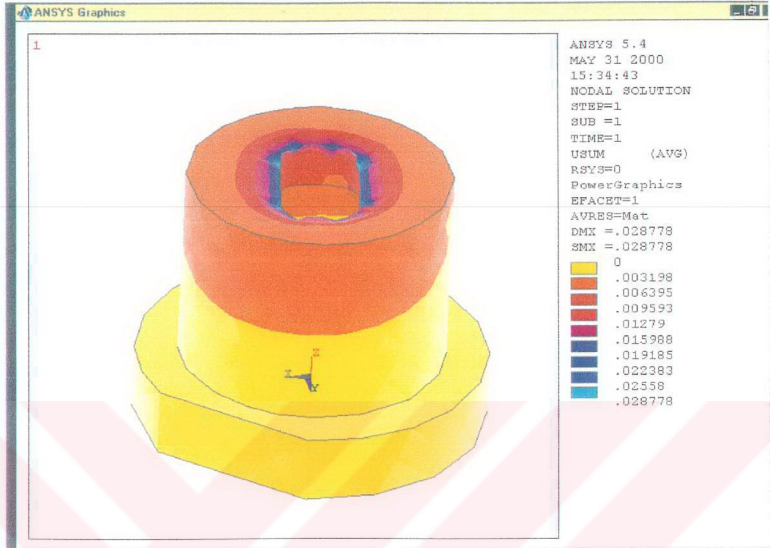
3.1 Analiz Sonuçlarının Sunumu Ve Sonuçların İrdelenmesi

Aşağıda verilmiş olan analiz sonuçlarını gösteren şekiller incelendiğinde (bkz şekil 3.1, Şekil 3.2, Şekil 3.3, Şekil 3.4, Şekil 3.5, Şekil 3.6, Şekil 3.7 ve Şekil 3.8) geometrilerin hepsinde birbirine yaklaşık sonuçların elde edildiği görülmektedir. Bunun nedeni geometri boyutlarının birbirlerinden çok farklı olmayışı, mührelerin temel geometrilerinin de birbiri ile aynı olmasıdır. Toplam deformasyon sonuçları incelendiğinde mührelerdeki maksimum deformasyon miktarlarının 0.015mm – 0.020 mm arasında değiştiği görülmektedir. Bu değerler, kesme işlemi için tehlike arz etmeyen değerler olduğundan deformasyon miktarları kalıbın çalışmasında herhangi bir sorun çıkarmayacaktır. Mührelerdeki deformasyon hareketlerinin karakteri incelendiğinde, defomasyonun alt bölgedeki bir çizgiden sonra sıfır değerine düştüğü görülmektedir.

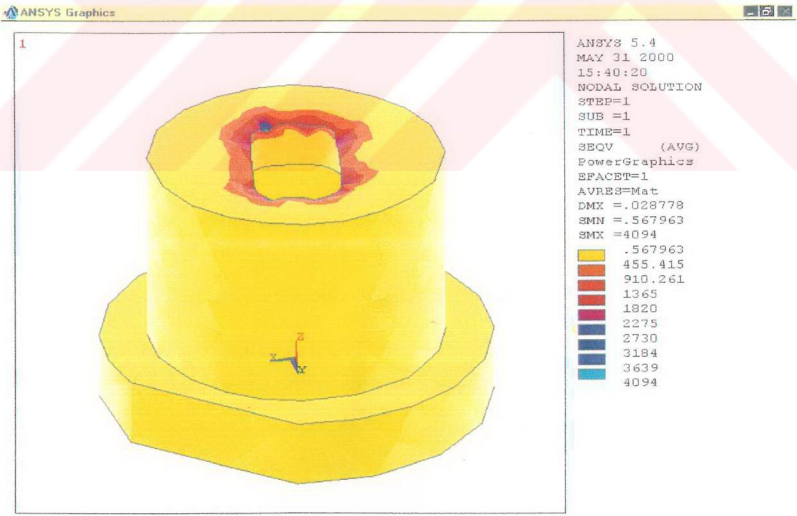
Mührelerin gerilme durumları incelendiğinde özellikle kesici ağız üzerinde gerilmelerin çok yüksek değerler aldığı gözlenmektedir. Ancak bu bölgelerde elde edilen bu çok yüksek değerlerin sebebi, o bölgelere noktasal yük uygulanmış olmasıdır. Gerçekte ise o bölgelerde çizgisel yayılı yük vardır. Bu nedenle çok yüksek gerilme değerleri gösteren noktadaki gerilme değerleri gerçekçi değildir. Süreklilik gösteren renk bantlarındaki değerler dikkate alınması gereken değerlerdir ve bu değerler de genelde $900 \text{ N/mm}^2 - 1300 \text{ N/mm}^2$ arasında değişen değerlerdir. Mühre malzemesi olarak kullanılan SVERKER21 takım çelikleri için 55 HRC sertlik değerinde verilen çekme dayanımı değeri 1700 N/mm^2 olduğundan (bkz. Çizelge 3.1.) bu bölgelerde kırılma meydana gelmeyecektir. Meydana gelen plastik deformasyon oranları da oldukça düşük olduğundan kalıp çalışmasında soruna yol açacak herhangi bir neden görülmemektedir.

Çizelge 3.1 SVERKER 21 Takım Çeliğinin özellikleri (Ganiyusufoğlu, 1999)

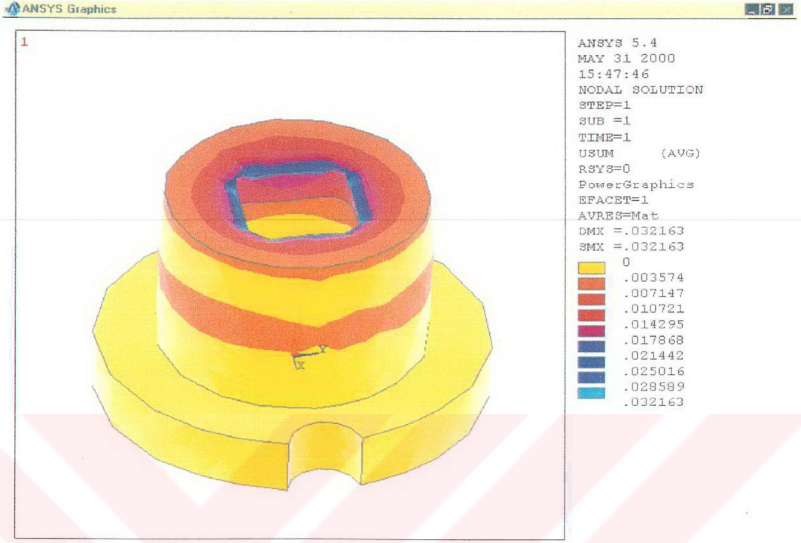
Malzeme Adı	DIN normu	%C	%Si	%Mn	%Cr	%Mo	%V
SVERKER 21	X 155 CrV Mo 121	1.55	0.30	0.40	11.80	0.80	0.80



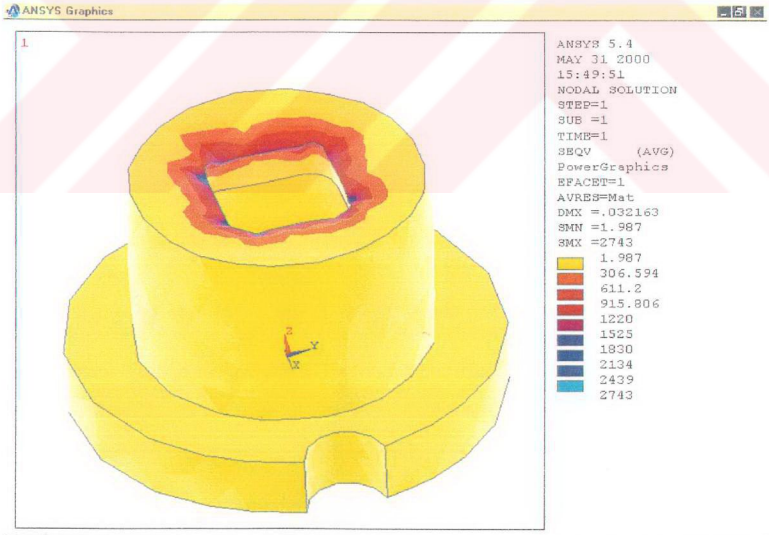
Şekil 3.1 Küçük dörtgen mühenin toplam deformasyon durumu.



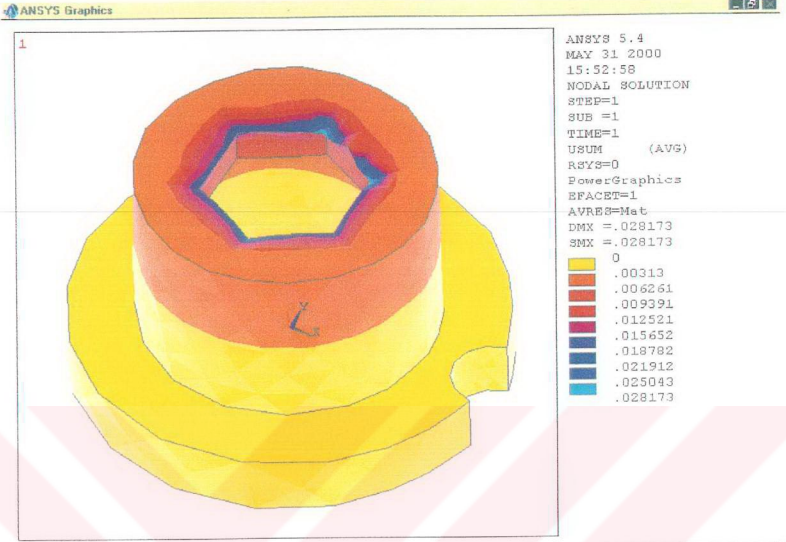
Şekil 3.2 Küçük dörtgen mühenin toplam gerilme durumu.



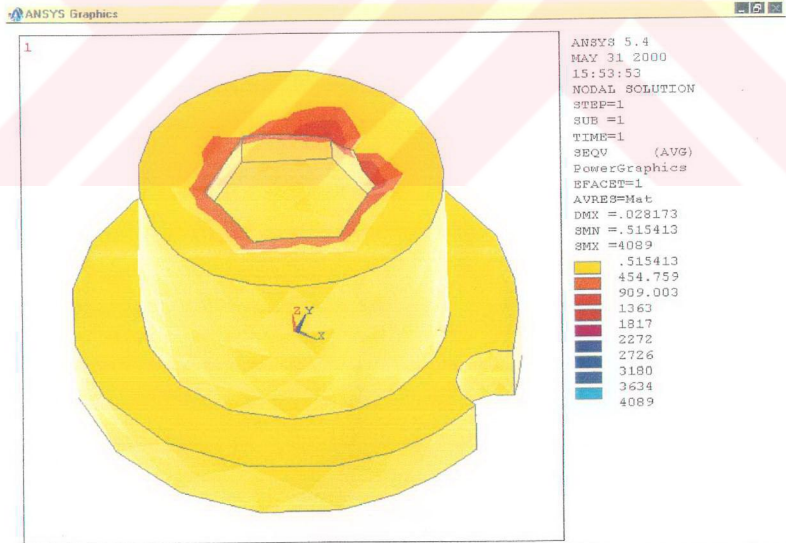
Şekil 3.3 Büyük dörtgen mührü toplam deformasyon durumu.



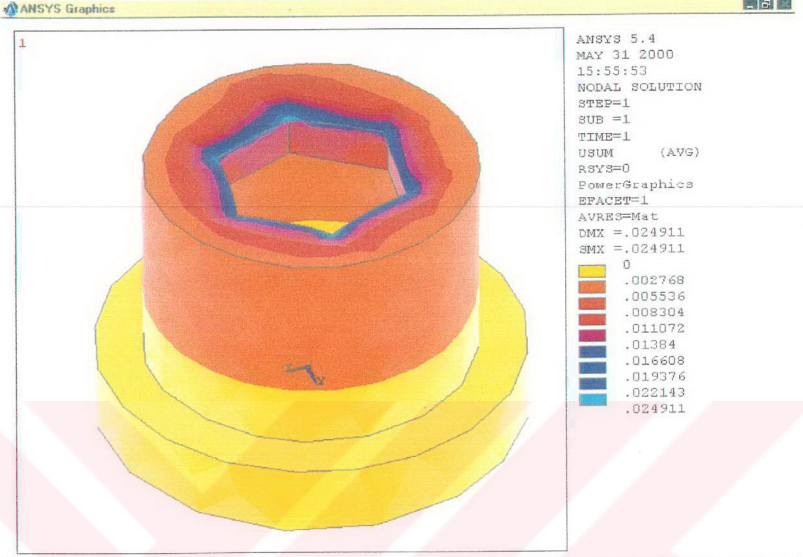
Şekil 3.4 Büyük dörtgen mührünün toplam gerilme durumu.



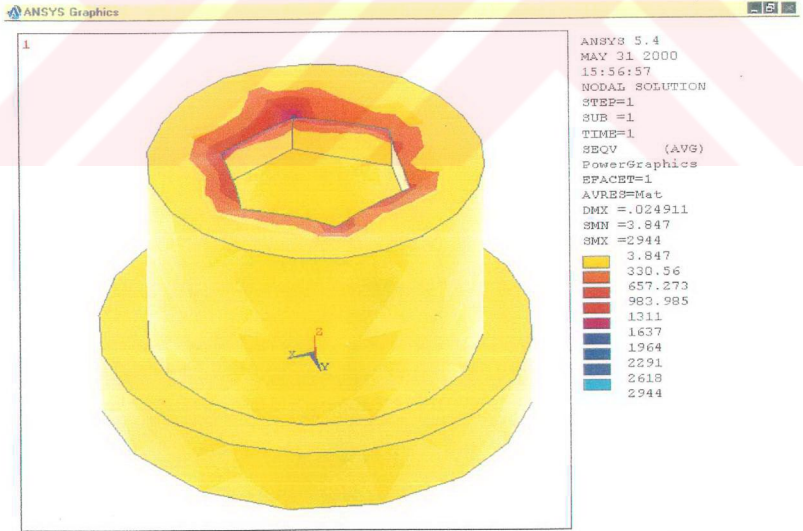
Şekil 3.5 Büyük altıgen mührü toplam deformasyon durumu.



Şekil 3.6 Büyük altıgen mührünün toplam gerilme durumu.



Şekil 3.7 Küçük altıgen mührü toplam deformasyon durumu.



Şekil 3.8 Küçük altıgen mührü toplam gerilme durumu.

3.2 CAD/CAM/CAE Kavramları Üzerine Yorum ve Öneriler

CAD/CAM işlemlerinin yapılması için kullanılan sistemde CAD ve CAM kısımları birbirine entegre çalışmıyorsa yani CAD programından CAD programına geometri transferi yapmak için farklı bir format kullanılıyorsa (IGES, STL, DXF gibi) transfer esnasında geometri yüzeylerinde kayıplar ve hatalar meydana gelebilir. Bu hataların meydana gelmesini engellemenin en uygun çözümü CAD/CAM modüllerini içerisinde barındıran sistemlerin ya da birbirine daha sonra entegre olabilen sistemlerin tercih edilmesidir.

CAD ortamında tasarım yapılırken şayet yüzey modelleme tekniği ile parçaların oluşturulması isteniyorsa, parça geometrilerinin yüzey modellemeye uygun olması gerekmektedir. Örneğin saç parçaların ya da diğer bir deyişle kalınlığı diğer boyutlarına göre ince olan parçaların tasarımı yüzey modelleme ile yapılabilir. Kalın kesitli parçaların tasarımı yüzey modelleme tekniği ile yapılmışsa, bu parçanın analizi için uygun eleman tipi bulunamayabilir ve parça analizinin yapılması için parçanın tekrar modellenmesi gerekebilir. Katı modelleme ile ise her tip parçanın tasarımı ve analizi yapılabilir.

Diğer boyutlarına göre kalınlığı ince olan parçaların analizinde katı (solid) elemanların yerine yüzey tipi elemanların (kabuk eleman, levha eleman) kullanımı, analiz süresinin kısaltılması bakımından çok kullanışlı olabilir. Buna göre ince geometrilerin tasarımının yüzey modelleme tekniği ile yapılması daha kullanışlı olmaktadır.

Sistemlerin tasarımı esnasında güvenilir bir tasarım yapabilmek amacı ile sonlu elemanlar analizi yapılarak tasarımların doğrulanması, güvenilir sonuçlar verecektir. Ancak bu tasarım süresini çok fazla uzatacaktır. Bu tip bir çözüm yerine, tasarım esnasında CAD sistemine entegre olabilen ve basit çözümlerden ibaret olan analiz programları (DESIGNSPACE gibi) tercih edilebilir. Bu tip programlarda geometrinin transfer edilmesi, alt elemanlara bölünmesi ve düğüm noktalarına yük ya da sınırlama uygulanması gibi vakit alıcı işlemler yoktur. Ayrıca çözüm süreleri de göreceli olarak daha kısadır. Ancak tasarımın bitmesinin ardından doğrulama yine de Sonlu Elemanlar Analizi yapılarak sağlanmalıdır. Zira bu tip basit programların verdiği sonuçlar, bir sonlu eleman analizine göre oldukça kaba değerlerdir.

KAYNAKLAR

SAS IP,Inc., (1996), “Ansys/Autofea 3D User’s Guide”, 1996, Houston

Bathe, J. K., “Finite Element Procedures In Engineering Analysis”, Prentice-Hall, Inc., 1982, Englewood Cliffs, New Jersey 07632

Cook, R. D., Malkus, D. S., Plesha, M. E., “Concepts And Applications Of Finite Element Analysis”, John Wiley And Sons Inc., 1989, New York

Dalođlu, A., “Sonlu Elemanlar Analizi’ne Giriř ”, KTÜ Ders Notları (yayımlanmamıř)

Ganiyusufođlu, M., “Takım elikleri El Kitabı”, Assab Korkmaz elik Tic. Ve San. A.ř. yayınları, 1999, İstanbul

Güneř, A. T., “Pres İřleri Tekniđi”, Makina Mühendisleri Odası Yayınları, 129, 1989, İstanbul

Openmind Software Technologies GmbH, “Hypermill V5.0 User’s Guide”, 1999, Unterföhring

Kurt, H., “Kalıpcılık Tekniđi Ve Konstrüksiyon”, Orkun Yayınevi, 1988, Sirkeci/İstanbul

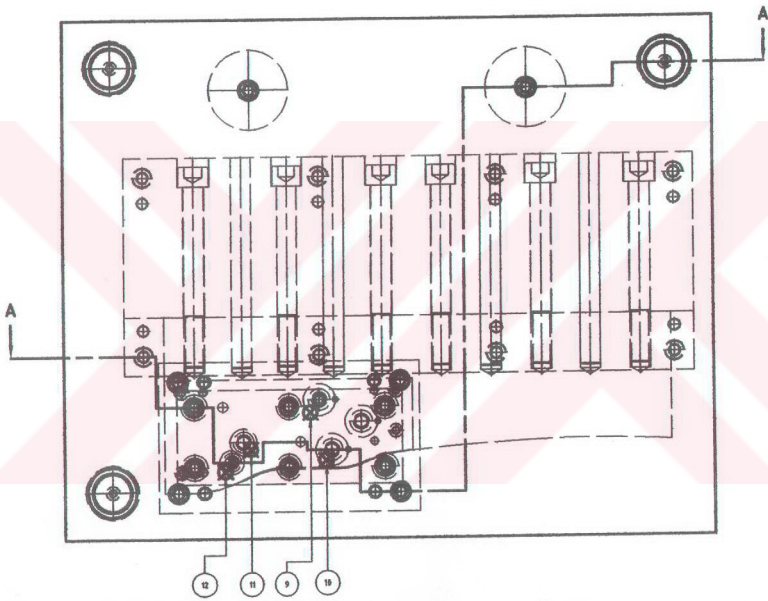
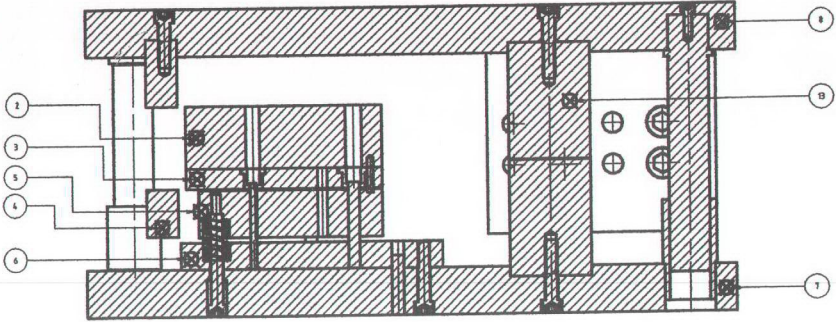
Autodesk Inc., “Mechanical Desktop R4 User’s Guide” , 1998, San Rafael

EKLER

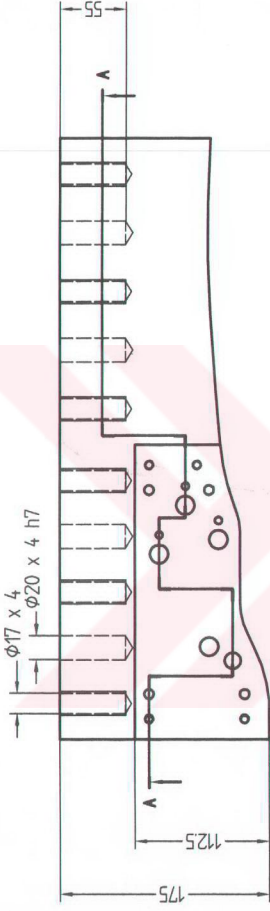
- Ek 1 Örnek Kalıp Projesinin Detay Resimleri
 - Ek 2 Örnek Analizde Kullanılan ELEMEN 92'nin özellikleri
 - Ek 3 Gaus Eliminasyon Metodunun FORTRAN Programlama Dili İle Yazılmış Kodları
 - Ek 4 C++ Programlama Dili İle Yazılmış Sonlu Elemanlar Analiz Programı Kodları
-



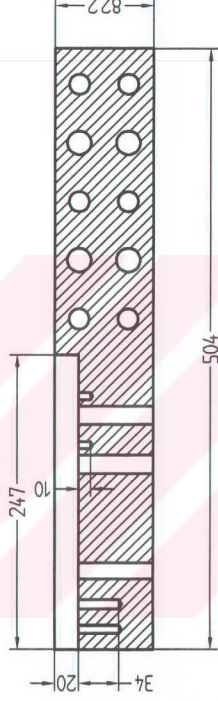
A-A KESİTİ



SIRA NO	DEĞİŞİKLİKLER			TARİH	İSİM	
Adet NO. 1	Tarih Date	İsim Name	Ölçek Scale	İz Düşüm Projection plane	Yüzey Finish	Malzeme Material
Cizen Drawn by	05/04/00	T.YAMAN	1 : 5	DIN 6 T1		-
Kontrol Checked			Genel tolerans Gn. tolerance	Ağırlık Weight	Ref. no. YC15 H107K08 AH	
Onay Approved			DIN 7168 m		Parça no. Part no. KL01-01	
bosai Bosai-Mimaysan AŞ		Parça adı :		KROS YAN DELİK KALIBI		

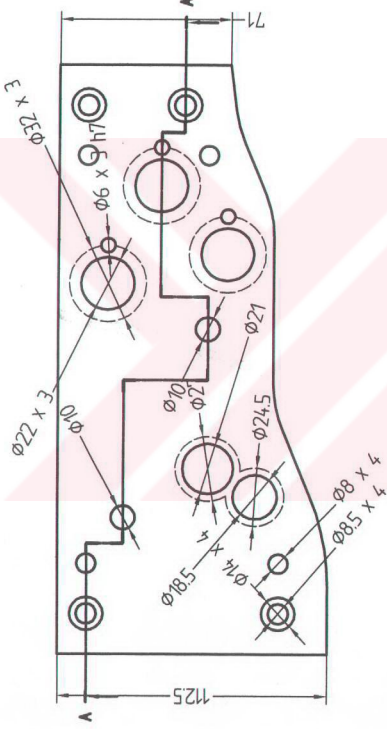


A-A KESİTİ



SIRA NO	DEĞİŞİKLİKLER	TARİH	İSİM
Adet NO: 1	İsim Name: T.YAMAN	Yüzey Finish	Malzeme Material: C1040
Çizim by: 05/04/00	Ölçek Scale: 1 : 4	izqisım Projection plane: DIN 6 T1	
Kontrol Checked	Genel tolerans Gn. tolerance	Ağırlık Weight	Ref. no. OT16
Onay Approved	DIN 7168 m		Parça no. KL01-02
Parça adı : ÇELİK DESTEK			

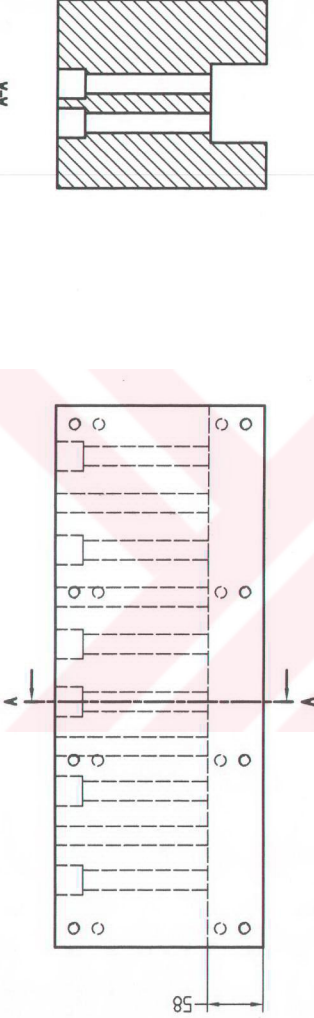
Bosai
Bosai-Mimaysan AŞ



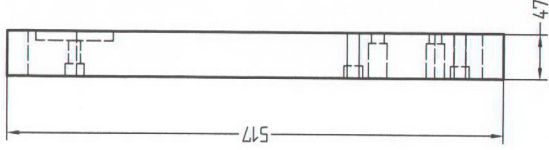
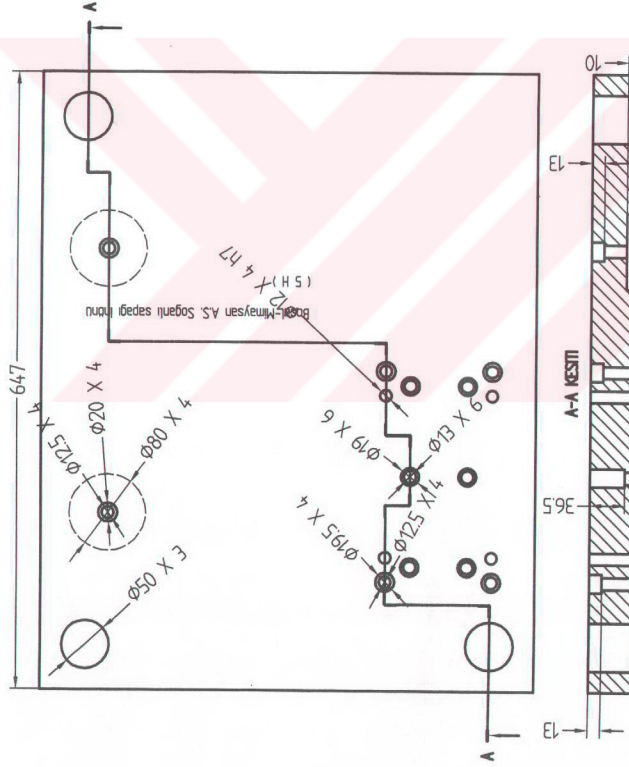
A-A KESİTİ

SIRA NO	DEĞİŞİKLİKLER	TARİH	İSİM
Adet NO: 1	İsmi Name: T.YAMAN	Yüzey Finish: SVERKER 21	Malzeme Material: SVERKER 21
Çizim by: 05/04/00	Ölçek Scale: 1 : 2	Projeksiyon plane: DIN 6 T1	Ref. no. Ref. no. OT16
Kontrol Checked:	Genel tolerans Gr. tolerance: DIN 7168 m	Ağırlık weight:	Parça no. Part. no. KL01-03
Onay Approved:			
Bosal Bosal-Mimaysan AŞ			Parça adı : DISI ÇELİK

A-A

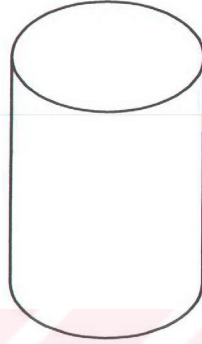
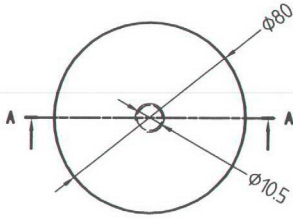


SIRA NO		DEĞİŞİKLİKLER				TARİH	İSİM
Adet	1	İzçisim	Ölçek	Projeleme	Yüzey	Malzeme	
Çizim	05/04/00	Scale	1 : 5	plane	Finish	C.1040	
Kontrol	by T.YAMAN	Drawn	Genel tolerans	⊕ DIN 6 T1	Ref. no.	0T16	
Checked		Drawn	Gen. tolerance	⊕ DIN 6 T1	Ref. no.	KL01-04	
Onay		Drawn	DIN 7168 m	Ağırlık	Ref. no.		
Approved		Drawn		Weight	Part no.		
Bosai Bosai-Mıymaysan AŞ							
Parça adı : DESTEK BLOĞU							

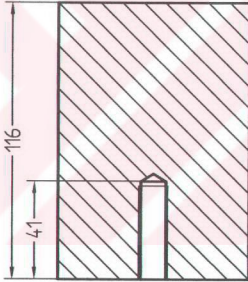


SIRA NO		DEĞİŞİKLİKLER				TARİH		İSİM	
Adet	1	Tarih	İsim	Ölçek	Projeksiyon	Yüzey	Malzeme	Ref. no.	Parça no.
NO	1	Date	Name	Scale	plane	Finish	Material	Ref. no.	Part no.
Çizen	05/04/00	T.YAMAN	1 : 5	DIN 6 T1	0116	ST37	0116	KL01-07	
Kontrol				Genel tolerans					
Çekilmiş				Gm. tolerance					
Onay				DIN 7168 m					
Approved									
Bosai								Parça adı :	
Bosal-Tirmaysan AŞ								USTPLAKA	

Bu tasarım , BMT 11 CAD-CAM tarafından yapılmıştır.

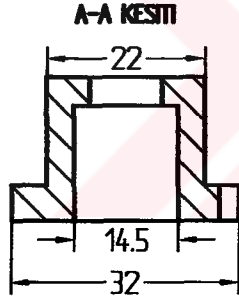
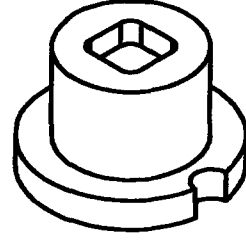
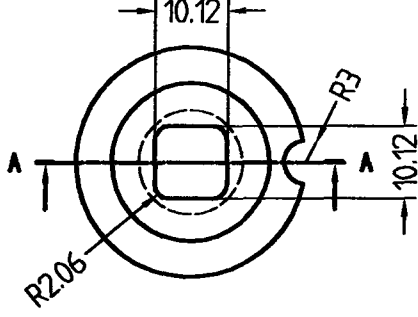


A-A KESİTİ



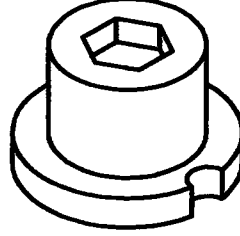
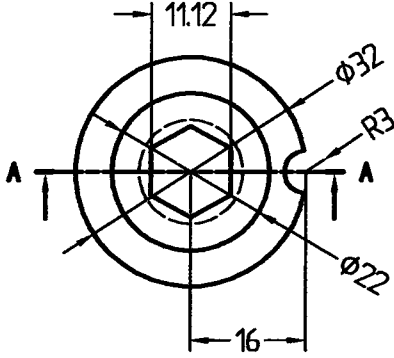
SIRA NO		DEĞİŞİKLİKLER				TARİH		İSİM	
Adet NO.	1	Tarih Date	İsim Name	Ölçek Scale	izdüşüm Projection plane	Yüzey Finish		Malzeme Material	
Cizen Drawn by	05/04/00	T.YAMAN	1 : 1	⊕ DIN 6 T1			C1040		
Kontrol Checked			Genel tolerans Gn. tolerance	Ağırlık Weight	Ref. no.	DT16			
Onay Approved			DIN 7168 m		Parça no. Part no.	KL01-08			
bosal		Parça adı :		STOPER					
Bosal-Mimaysan AŞ									

Bu tasarım , BMT 11 CAD-CAM tarafından yapılmıştır.

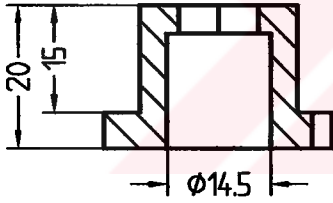


SIRA NO		DEĞİŞİKLİKLER			TARİH	İSİM	
Adet NO.	1	Tarih Date	İsim Name	Ölçek Scale	izdüşüm Projection plane	Yüzey Finish	Malzeme Material
Cizen Drawn by	05/04/00	T.YAMAN	1 : 1	⊕ DIN 6 T1			SVERKER 21
Kontrol Checked			Genel tolerans Gn. tolerance	Ağırlık Weight	Ref. no.	DT16	
Onay Approved			DIN 7168 m		Parça no. Part no.	KL01-09	
bosal Bosal-Mimaysan AŞ		Parça adı : BUYUK DORTGEN MUHRE					

Bu tasarım , BMT 11 CAD-CAM tarafından yapılmıştır.

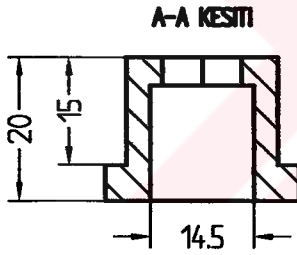
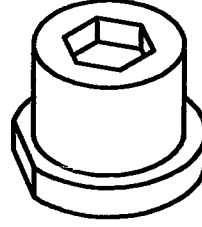
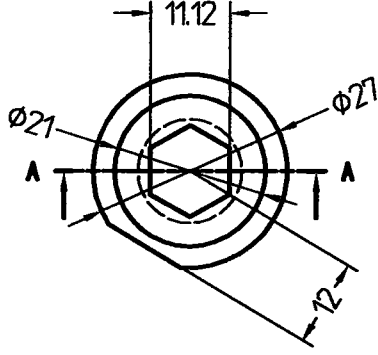


A-A KESİTİ



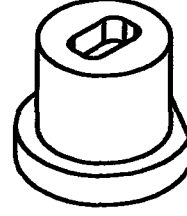
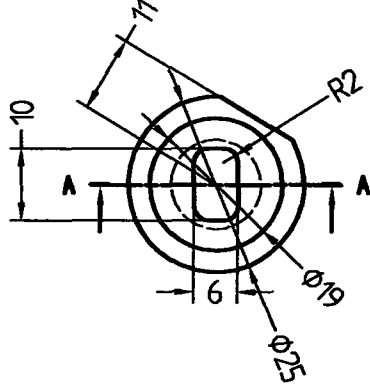
SIRA NO		DEĞİŞİKLİKLER			TARİH	İSİM	
Adet NO.	2	Tarih Date	İsim Name	Ölçek Scale	izdüşüm Projection plane	Yüzey Finish	Malzeme Material
Çizen Drawn by	05/04/00	T.YAMAN	1 : 1	⊕ DIN 6 T1			SVERKER 21
Kontrol Checked			Genel tolerans Gn. tolerance	Ağırlık Weight	Ref. no.	DT16	
Onay Approved			DIN 7168 m		Parça no. Part no.	KL01-10	
bosal Bosal-Mimaysan AŞ		Parça adı : BUYUK ALTIGEN MUHRE					

Bu tasarım , BMT 11 CAD-CAM tarafından yapılmıştır.

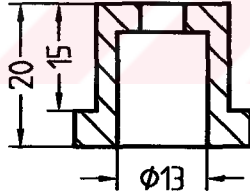


SIRA NO		DEĞİŞİKLİKLER			TARİH		İSİM	
Adet NO.	1	Tarih Date	İsim Name	Ölçek Scale	izdüşüm Projection plane	Yüzey Finish	Malzeme Material	
Çizen Drawn by	05/04/00	T.YAMAN	1 : 1	⊕ DIN 6 T1			SVERKER 21	
Kontrol Checked			Genel tolerans Gn. tolerance	Ağırlık Weight	Ref. no.	OT16		
Onay Approved			DIN 7168 m		Parça no. Part no.	KL01-11		
bosal Bosal-Mimaysan AŞ		Parça adı : KUCUK ALTIGEN MUHRE						

Bu tasarım , BMT 11 CAD-CAM tarafından yapılmıştır.



A-A KESİTİ



SIRA NO		DEĞİŞİKLİKLER			TARİH		İSİM	
Adet NO.	1	Tarih Date	İsim Name	Ölçek Scale	İz düşüm Projection plane	Yüzey Finish	Malzeme Material	
Cizen Drawn by	05/04/00	T.YAMAN	1 : 1	⊕ DIN 6 T1			SVERKER 21	
Kontrol Checked			Genel tolerans Gn. tolerance	Ağırlık Weight	Ref. no.	OT16		
Onay Approved			DIN 7168 m		Parça no. Part no.	KL01-12		
bosal Bosal-Mıymaysan AŞ		Parça adı : KUCUK DORTGEN MUHRE						

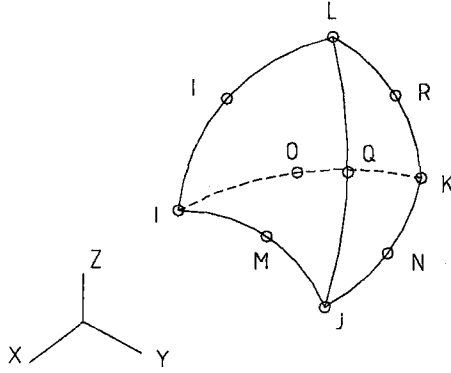
EK 2. ELEMAN 92

“Solid 92” elemanı kuadratik yerdeğiřtirme davranıřına sahiptir ve düzensiz ađ yapısına sahip sistemleri modellemek için (genellikle farklı CAD/CAM sistemlerinden elde edilen geometriler için) çok uygundur. Bu elemanın alternatif elemanları 20 düđüm noktalı SOLID95 ve 4 düđüm noktalı tetrahedral SOLID72 elemanıdır.

Bu eleman, herbir düđüm noktasında x, y ve z yönlerinde yerdeğiřtirme serbestlik derecelerine sahiptir. Bu elemanın ek olarak plastisite, büyük Őekil deđiřtirebilme ve pekleřme özellikleri de vardır.

Girdi Verileri:

Őekil Ek1.1’ de düđüm noktalarının konumları ve bu eleman için koordinat sistemi gösterilmektedir. Bu elemana basınç yüklemesi yapılmak istenirse bu yükleme tipi yüzey yükleri olarak tanımlanmalıdır. Sıcaklık ya da akı deđerleri ise elemana hacimsel yükler olarak verilmelidir. Elemanın girdi veri tipleri Çizelge Ek 1. 1’ de açıklanmıřtır.



Őekil Ek 1.1 Solid 92 Eleman yapısı

Çıktı Verileri:

Eleman iki tip çıktı bilgisi verebilmektedir: 1) düğüm noktalarına göre ve 2) elemanlara göre. Bu elemanın verdiği çıktı veri bilgileri Çizelge Ek 1.2' de verilmektedir. Çizelgedeki "O" harfi, söz konusu verinin çıktı dosyasında mevcut olabildiğini (ANSYS'de çözülen ve başka bir postprosesörde incelemek için), "R" harfi ise söz konusu bilginin ANSYS ortamından okunabileceği anlamına gelmektedir.

Elemanın getirdiği sınırlamalar:

Bu eleman sadece katı modellere uygulanabilmektedir. Modelin hacmi sıfır (sadece yüzeyle ibaret model) olamaz.

bu elemanda ek olarak gerilme sertleşmesi (pekleşme) özelliği vardır.

Çizelge Ek 1.1 Solid 92 elemanın girdi veri tipleri

ELEMAN ADI	SOLID 92
DÜĞÜM NOKTALARI	I, J, K, L, M, N, O, P, Q, R
SERBESTLİK DERECELERİ	UX, UY, UZ
EK SINIRLAMA	GEREK YOK
MALZEME ÖZELLİKLERİ	EX, EY, EZ, ALPX, ALPY, ALPZ (PRXY, PRYZ, PRXZ ya da NUXY, NUYZ, NUXZ), DENS, GXY, GYZ, GXZ, DAMP
YÜZEY YÜKLEMELERİ	(J-I-K) yüzeyi, (I-J-L) yüzeyi, (J-K-L) yüzeyi ve (K-I-L) yüzeyi
HACİMSSEL YÜKLER	Sıcaklıklar: T(I), T(J), T(K), T(L), T(M), T(N), T(O), T(P), T(Q), T(R) Akı değerleri: FL(I), FL(J), FL(K), FL(L), FL(M), FL(N), FL(O), FL(P), FL(Q), FL(R),
EK ÖZELLİKLER	Plastisite, Büyük Yerdeğiştirme, Gerilme Sertleşmesi

Çizelge Ek 1.2 Solid 92 Elemanın Çıktı Veri Tipleri

İSİM	AÇIKLAMA	“O”	“R”
EL	Eleman Numarası	•	•
NODES	Kenar Düğüm Noktaları (I, J, K, L)	•	•
MAT	Malzeme Numarası	•	•
VOLU:	Hacim	•	•
CEN:X,Y,Z	Kütle Merkezi X,Y,Z koordinatları	•	•
PRES	Düğüm Noktası Basınç Değerleri	•	•
TEMP	Düğüm Noktası Sıcaklık Değerleri	•	•
FLUEN	Düğüm Noktası Akış Değerleri	•	•
S:EQV	Eşdeğer Gerilme	•	•
S:X,Y,Z,XY,YZ,XZ	Gerilmeler	•	•
S:1,2,3	Asal Gerilmeler	•	•
EPEL:X,Y,Z,XY,YZ,XZ	Elastik Deformasyonlar	•	•
EPEL: 1,2,3	Asal Eksenlerdeki Elastik Deformasyonlar	•	•
EPPL:X,Y,Z,XY,YZ,XZ	Plastik Deformasyonlar	•	•

EK 3. GAUS ELİMİNASYON METODUNUN FORTRAN PROGRAMLAMA DİLİ PROGRAM KODLARI

DIMENSION A (NWK) , V (1) , MAXA (1)

C PERFORM L * D * L (T) FACTORISATION OF STIFFNESS MATRIX

IF (KKK - 2) 40 , 150 , 153

40 DO 140 N = 1 , NN

KN = MAXA (N)

KL = KN + 1

KU = MAXA (N + 1) - 1

KH = KU - KL

IF (KH) 110 , 90 , 50

50 K = N = KH

IC = C

KLT = KU

DO 80 J = 1 , KH

IC = IC + 1

KLT = KLT - 1

KI = MAXA (K)

ND = MAXA (K + 1) - KT - 1

IF (ND) 80 , 80 , 60

60 KK = MIN0 (IC , ND)

C = 0.

DO 70 L = 1 , KK

70 C = C + A (KI + L) * A (KLT + L)

A (KLT) = A (KLT) - C

80 K = K + 1

90 K = N

B = 0.

DO 100 KK = KL , KU

K = K - 1

KI = MAXA (K)

C = A (KK) / A (KI)

```

      B = B + C * A (KK)
100 A (KK) = C
      A (KN) = A (KN) - E
110 IF (A (KN)) 120 , 120 , 140
120 WRITE (IOUT , 2000) N , A (KN)
      STOP
140 CONTINUE
      RETURN

```

C REDUCE RIGHT HAND SIDE LOAD VEKTOR

```

150 DO 180 N = 1 , NN
      KL = MAXA (N) + 1
      KU = MAXA (N + 1) - 1
      IF (KU - KL) 180 , 160 , 160
160 K = N
      C = 0.
      DO 170 KK = KL , KU
      K = K - 1
170 C = C + A (KK) * V (K)
      V (N) = V (N) - C
180 CONTINUE

```

C BACK SUBSTITUTE

```

      DO 200 N = 1 , NN
      K = MAXA (N)
200 V (N) = V (N) / A (K)
      IF (NN .EQ. 1) RETURN
      N = NN
      DO 230 L = 2 , NN
      KL = MAXA (N) + 1
      KU = MAXA (N + 1) - 1

```

```
IF (KU - KL) 230, 210, 210
210 K = N
DO 220 KK = KL, KU
K = K - 1
220 V (K) = V (K) - A (KK) * V (N)
230 N = N - 1
RETURN
2000 FORMAT (//48H STOP - STIFFNESS MATRIX POSITIVE DEFINITE ,//
1          32H NONPOSITIVE PIVOT FOR EQUATION ,I4,//
2          10H PIVOT = ,E20.12 )
END
```



EK 4. C++ PROGRAMLAMA DİLİ İLE YAZILMIŞ SONLU ELEMAN ANALİZ PROGRAMI KODLARI

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

static int T;
static float a11, a66, a12, a26, a16, a22, K[3][3][5][5], KTOU[3][3][6][6], e;
static float KTUT[3][3][6][6], KTOT[3][3][6][6], KTOA[3][3][6][6];
FILE *mtp;

void AHESAP(void);
void INTEGRAL( float ALPHA, float BETA);
void yazdir1(void);
void hesapKTOU(void);
void hesapKTUT(void);
void hesapKTOT(void);
void hesapKTOA(void);
void ADMson(void);
void UMson(void);
void JDson(void);
void forceson(float *);
void main(void)

{

int cevap, LN, HN ;
float L, H, ALPHA, BETA;

printf("kesit genisligini giriniz:");
scanf("%f",&L);
fflush(stdin);
printf("kesit yuksekligini giriniz:");
scanf("%f",&H);
fflush(stdin);
printf("genislik boyunca eleman sayisini giriniz:");
scanf("%d",&LN);
fflush(stdin);
printf("yuksekluk boyunca eleman sayisini giriniz:");
scanf("%d",&HN);
fflush(stdin);
printf("E oranini giriniz:");
scanf("%f",&e);
fflush(stdin);
T= (HN+1)*(LN+1);

```

```

BETA = L/LN/2.0;
ALPHA = H/HN/2.0;
AHESAP();
INTEGRAL(ALPHA, BETA);
printf("yerel katilik matrislerinin ekrana yazilmasini istermisiniz?(E/H):");
cevap = getch();
printf("\n");

if (cevap == 'E')
    yazdir1();

if (cevap == 'e')
    yazdir1();
else
    ;
hesapKTOU();
hesapKTUT();
hesapKTOT();
hesapKTOA();
printf("sinir kosullari uygulandi\n");
ADMson();
UMson();
JDson();
forceson(&BETA);
printf("\n");
getch();
}

void AHESAP (void)

{

float p1, p2, et1, et2, em1, em2, el1, el2, elm1;

p1=0.3;
p2=0.3;
et1=0.5;
et2=0.5;
em1= 1.0/(2.0*(1.0 + p1));
em2= e/(2.0*(1.0 + p2));
el1 = p1/((1.+p1)*(1.-.*p1));
el2 = e*p2/((1.+p2)*(1.-.*p2));
a11 = em1 * et1+em2*et2+(el1+em1)*et1+(el2+em2)*et2;
elm1 = (el1+2.*em1)*et2+(el2+2.*em2)*et1;
a11 = a11-et1*et2*(el1-el2)*(el1-el2)/elm1;
a12 = el1*et1+el2*et2;
a12 = a12-et1*et2*((el1 + 2. * em1) - (el2 + 2. * em2)) / elm1;
a22 = (el1 + 2. * em1) * et1 + (el2 + 2. * em2)*et2;

```

```

a22 = a22 - et1 * et2 * (((el1 + 2. * em1) - (el2 + em2 * 2.)) * ((el1 + 2. * em1) - (el2 + em2 *
2.))) / elm1;
a66 = em1 * em2 / (em1 * et2 + em2 * et1);
a16 = 0.;
a26 = 0.;
printf("a11=%f\n", a11);
printf("a16=%f\n", a16);
printf("a66=%f\n", a66);
printf("a12=%f\n", a12);
printf("a26=%f\n", a26);
printf("a22=%f", a22);
}

```

```

void INTEGRAL(float ALPHA, float BETA)

```

```

{
/* 3 nokta ve ağırlık kullanılarak Gaus – Legende nümerik integrasyonun yapıldığı alt
fonksiyon*/
/* Integrasyonun yapılması ardından takip eden döngüde yerel eleman katılık matrislerinin
hesaplandığı alt fonksiyon*/

float BDB11, BDB12, BDB21, BDB22, BDB11S, BDB12S, BDB21S, BDB22S;

float ETT[5], KSS[5], r[4], ALPH[4];

int i, j, k, m;

r[1] = 0.77459;
r[2] = 0.00000;
r[3] = -0.77459;

ALPH[1] = 0.55555;
ALPH[2] = 0.88888;
ALPH[3] = 0.55555;

BDB11 = 0;
BDB12 = 0;
BDB21 = 0;
BDB22 = 0;
BDB11S = 0;
BDB12S = 0;
BDB21S = 0;
BDB22S = 0;

for(i=1; i<5; ++i){

```

```
for(j=1; j<5; ++j){
```

```

  BDB11 = 0;
  BDB12 = 0;
  BDB21 = 0;
  BDB22 = 0;
  BDB11S = 0;
  BDB12S = 0;
  BDB21S = 0;
  BDB22S = 0;
```

```
for(k = 1; k<4; ++k){
```

```

  ETT[1] = -(1./4/ALPHA) + (1./4/ALPHA) * r[k];
  ETT[2] = -(1./4/ALPHA) - (1./4/ALPHA) * r[k];
  ETT[3] = (1./4/ALPHA) + (1./4/ALPHA) * r[k];
  ETT[4] = (1./4/ALPHA) - (1./4/ALPHA) * r[k];
  BDB11 = 0;
  BDB12 = 0;
  BDB21 = 0;
  BDB22 = 0;
```

```
for(m = 1; m<4; ++m){
```

```

  KSS[1] = -(1./4/BETA) + (1./4/BETA) * r[m];
  KSS[2] = (1./4/BETA) - (1./4/BETA) * r[m];
  KSS[3] = (1./4/BETA) + (1./4/BETA) * r[m];
  KSS[4] = -(1./4/BETA) - (1./4/BETA) * r[m];
```

```

  BDB11 = BDB11 + (((((a11 * KSS[j] + a16 * ETT[j]) * KSS[i] +
a66 * ETT[j]) * ETT[i]))) * ALPH[m]);
```

```

  BDB12 = BDB12 + (((((a12 * ETT[j] + a16 * KSS[j]) * KSS[i] +
a66 * KSS[j]) * ETT[i]))) * ALPH[m]);
```

```

  BDB21 = BDB21 + (((((a12 * KSS[j] + a26 * ETT[j]) * ETT[i] +
a66 * ETT[j]) * KSS[i]))) * ALPH[m]);
```

```

  BDB22 = BDB22 + (((((a22 * ETT[j] + a26 * KSS[j]) * ETT[i] +
a66 * KSS[j]) * KSS[i]))) * ALPH[m]);
```

```
}
```

```
  BDB11S = BDB11S + (BDB11 * ALPHA[k]);
```

```
  BDB12S = BDB12S + (BDB12 * ALPHA[k]);
```

```
  BDB21S = BDB21S + (BDB21 * ALPHA[k]);
```

```
  BDB22S = BDB22S + (BDB22 * ALPHA[k]);
```

```
}
```

```
K[1][1][i][j] = BDB11S * ALPHA * BETA;
```

```
K[1][2][i][j] = BDB12S * ALPHA * BETA;
```

```
K[2][1][i][j] = BDB21S * ALPHA * BETA;
```

```
K[2][2][i][j] = BDB22S * ALPHA * BETA;
```

```

    }
}
}

void yazdir1(void)
{
/* Kullanıcı isteğine bağlı olarak yerel eleman katılık matrislerinin ekrana yazdırılması işleminin
yapıldığı alt fonksiyon*/

int s=0;
int x=1;
int i, j;

for(i=1; i<5; ++i){

printf("-----");

for(j=1; j<5; ++j){
printf("\n");
printf("K(1 1 %d %d) = %f\n", i, j, K[1][1][i][j]);
printf("K(1 2 %d %d) = %f\n", i, j, K[1][2][i][j]);
printf("K(2 1 %d %d) = %f\n", i, j, K[2][1][i][j]);
printf("K(2 2 %d %d) = %f\n", i, j, K[2][2][i][j]);
++s;

if(s==4*x){
++x;
printf("\n");
puts("devam etmek icin bir tusa basiniz");
getch();
}

}

}

}

void hesapKTOU(void)
{

int i, j;

for(i=1; i<3; ++i){

```

```

for(j=1; j<3; ++j){
    KTOU[i][j][1][1] = K[i][j][1][1];
    KTOU[i][j][1][2] = K[i][j][1][4];
    KTOU[i][j][1][3] = 0;
    KTOU[i][j][2][2] = K[i][j][4][4] + K[i][j][1][1];
    KTOU[i][j][5][5] = K[i][j][4][4];
}
}

for(i=1; i<3; ++i){
    for(j=1; j<3; ++j){
        KTOU[i][j][2][3] = KTOU[i][j][1][2];
        KTOU[i][j][3][4] = KTOU[i][j][1][2];
        KTOU[i][j][4][5] = KTOU[i][j][1][2];
        KTOU[i][j][1][3] = 0;
        KTOU[i][j][1][4] = KTOU[i][j][1][3];
        KTOU[i][j][1][5] = KTOU[i][j][1][3];
        KTOU[i][j][2][5] = KTOU[i][j][1][3];
        KTOU[i][j][3][5] = KTOU[i][j][1][3];
        KTOU[i][j][3][3] = KTOU[i][j][2][2];
        KTOU[i][j][4][4] = KTOU[i][j][2][2];
    }
}

void hesapKTUT(void)
{
    int i, j;

    for(i=1; i<3; ++i){
        for(j=1; j<3; ++j){
            KTUT[i][j][1][1] = K[i][j][1][2];
            KTUT[i][j][1][2] = K[i][j][1][3];
            KTUT[i][j][1][3] = 0;
            KTUT[i][j][2][1] = K[i][j][4][2];
            KTUT[i][j][2][2] = K[i][j][4][3] + K[i][j][1][2];
            KTUT[i][j][5][5] = K[i][j][4][3];
        }
    }

    for(i=1; i<3; ++i){

```

```

for(j=1; j<3; ++j){
    KTUT[i][j][2][3] = KTUT[i][j][1][2];
    KTUT[i][j][3][4] = KTUT[i][j][1][2];
    KTUT[i][j][4][5] = KTUT[i][j][1][2];
    KTUT[i][j][1][4] = KTUT[i][j][1][3];
    KTUT[i][j][2][4] = KTUT[i][j][1][3];
    KTUT[i][j][2][5] = KTUT[i][j][1][3];
    KTUT[i][j][3][1] = KTUT[i][j][1][3];
    KTUT[i][j][3][5] = KTUT[i][j][1][3];
    KTUT[i][j][4][1] = KTUT[i][j][1][3];
    KTUT[i][j][4][2] = KTUT[i][j][1][3];
    KTUT[i][j][5][1] = KTUT[i][j][1][3];
    KTUT[i][j][5][2] = KTUT[i][j][1][3];
    KTUT[i][j][5][3] = KTUT[i][j][1][3];
    KTUT[i][j][3][3] = KTUT[i][j][2][2];
    KTUT[i][j][4][4] = KTUT[i][j][2][2];
    KTUT[i][j][3][2] = KTUT[i][j][2][1];
    KTUT[i][j][4][3] = KTUT[i][j][2][1];
    KTUT[i][j][5][4] = KTUT[i][j][2][1];
}
}
}

void hesapKTOT(void)
{
    int i, j;

    for(i=1; i<3; ++i){
        for(j=1; j<3; ++j){
            KTOT[i][j][1][1] = K[i][j][1][1] + K[i][j][2][2];
            KTOT[i][j][1][2] = K[i][j][1][4] + K[i][j][2][3];
            KTOT[i][j][1][3] = 0;
            KTOT[i][j][2][2] = K[i][j][1][1] + K[i][j][2][2] + K[i][j][3][3] + K[i][j][4][4];
            KTOT[i][j][5][5] = K[i][j][3][3] + K[i][j][4][4];
        }
    }

    for(i=1; i<3; ++i){
        for(j=1; j<3; ++j){
            KTOT[i][j][2][3] = KTOT[i][j][1][2];
            KTOT[i][j][3][4] = KTOT[i][j][1][2];
            KTOT[i][j][4][5] = KTOT[i][j][1][2];
            KTOT[i][j][1][4] = KTOT[i][j][1][3];

```

```

    KTOT[i][j][1][5] = KTOT[i][j][1][3];
    KTOT[i][j][2][4] = KTOT[i][j][1][3];
    KTOT[i][j][2][5] = KTOT[i][j][1][3];
    KTOT[i][j][1][5] = KTOT[i][j][1][3];
    KTOT[i][j][3][3] = KTOT[i][j][2][2];
    KTOT[i][j][4][4] = KTOT[i][j][2][2];
}
}
}

void hesapKTOA(void)
{
    int i, j;

    for(i=1; i<3; ++i){

        for(j=1; j<3; ++j){
            KTOA[i][j][1][1] = K[i][j][2][2];
            KTOA[i][j][1][2] = K[i][j][2][3];
            KTOA[i][j][2][2] = K[i][j][2][2] + K[i][j][3][3];
            KTOA[i][j][1][3] = 0;
            KTOA[i][j][5][5] = K[i][j][3][3];
        }
    }

    for(i=1; i<3; ++i){

        for(j=1; j<3; ++j){
            KTOA[i][j][2][3] = KTOA[i][j][1][2];
            KTOA[i][j][3][4] = KTOA[i][j][1][2];
            KTOA[i][j][4][5] = KTOA[i][j][1][2];
            KTOA[i][j][3][3] = KTOA[i][j][2][2];
            KTOA[i][j][4][4] = KTOA[i][j][2][2];
            KTOA[i][j][1][4] = KTOA[i][j][1][3];
            KTOA[i][j][1][5] = KTOA[i][j][1][3];
            KTOA[i][j][2][4] = KTOA[i][j][1][3];
            KTOA[i][j][2][5] = KTOA[i][j][1][3];
            KTOA[i][j][3][5] = KTOA[i][j][1][3];
        }
    }
}

```

```

void ADMson(void)
{
    int i, j, p;
    int m=1;
    FILE *adpp;

    adpp = fopen("adson.dat", "w");
    if(adpp==NULL){
        puts("dosya açılma hatası");
        exit(1);
    }
    printf("ad değerleri yazdırılıyor...\n");

    for(p=1; p<(T/5-2)+1; ++p){

        for(j=1; j<6; ++j){
            for(i=1; i<3; ++i){
                fprintf(adpp, "%f\n", KTOT[i][j][j]);
                ++m;
            }
        }
    }

    fclose(adpp);
    printf("AD degerleri adson.dat adli dosyaya yazdirildi\n");
    printf("ad dizisinin boyutu = %d\n",m-1);
}

```

```

void UMson(void)
{
    /* Genel katılık matrisinin birleştirilmesi işleminin yapıldığı alt fonksiyon*/

```

```

    int i, j, t, c, m, n;
    float KT[3][3][11][6];
    FILE *umpp;

    umpp = fopen("umson.dat", "w");
    if(umpp==NULL){
        puts("dosya açılma hatası");
        exit(1);
    }

    printf("U dizisi yazdırılıyor...\n");
    c=1;
    t=1;

```

```

for(j=2; j<6; ++j){
    i=j-1;
    fprintf(umpp, "%f\n", KTOT[1][2][t][t]);
    ++t;
    ++c;
    for(n=1; n<3; ++n){

        for(m=1; m<3; ++m){
            fprintf(umpp, "%f\n", KTOT[m][n][i][j]);
            ++c;
        }

    }

}
fprintf(umpp, "%f\n",KTOT[1][2][t][t]);

for(i=1; i<6; ++i){

    for(j=1; j<6; ++j){

        for(m=1; m<3; ++m){

            for(n=1; n<3; ++n){
                KT[m][n][i][j] = KTUT[m][n][i][j];
                KT[m][n][i+5][j] = KTOT[m][n][i][j];
            }

        }

    }

}

c=22;
for(t=1; t<(T/5-3)+1; ++t){

    for(j=1; j<2; ++j){

        for(n=1; n<3; ++n){

            for(i=1; i<6; ++i){

                for(m=1; m<3; ++m){

                    fprintf(umpp, "%f\n", KT[m][n][i][j]);
                    ++c;
                }

            }

        }

    }

}

```

```

    }
}
}
fprintf(umpp, "%f\n", KT[1][2][6][1]);
++c;
for(j=2; j<3; ++j){
    for(n=1; n<3; ++n){
        for(i=1; i<7; ++i){
            for(m=1; m<3; ++m){
                fprintf(umpp, "%f\n", KT[m][n][i][j]);
                ++c;
            }
        }
    }
}
fprintf(umpp, "%f\n", KT[1][2][7][2]);
++c;
for(j=3; j<4; ++j){
    for(n=1; n<3; ++n){
        for(i=2; i<8; ++i){
            for(m=1; m<3; ++m){
                fprintf(umpp, "%f\n", KT[m][n][i][j]);
                ++c;
            }
        }
    }
}
fprintf(umpp, "%f\n", KT[1][2][8][3]);
++c;
for(j=4; j<5; ++j){

```

```

for(n=1; n<3; ++n){
    for(i=3; i<9; ++i){
        for(m=1; m<3; ++m){
            fprintf(umpp, "%f\n", KT[m][n][i][j]);
            ++c;
        }
    }
}

}
fprintf(umpp, "%f\n", KT[1][2][9][4]);
++c;
for(j=5; j<6; ++j){
    for(n=1; n<3; ++n){
        for(i=4; i<10; ++i){
            for(m=1; m<3; ++m){
                fprintf(umpp, "%f\n", KT[m][n][i][j]);
                ++c;
            }
        }
    }
}

}

fprintf(umpp, "%f\n", KT[1][2][10][5]);
++c;
}

printf("U dizisi umson.dat isimli dosyaya yazdirildi\n");
printf("umson dizisinin boyutu = %d\n",c-1);
fclose(umpp);
}

void JDson(void)
{
    int i, SN, k, t;
    int *jd;
    FILE *jdpp;
}

```

```

jdpp=fopen("jdson.dat","w");
if(jdpp == NULL){
    puts("Dosya acilis hatasi!");
    exit(1);
}
SN = (T/5-2)*10+1;
jd = (int *)malloc(sizeof(int)*SN);
if(jd == NULL){
    printf("yetersiz bellek...\n");
    exit(1);
}

printf("jd dizisi hesaplaniyor...\n");
jd[1] = 0;
jd[2] = 1;
for(i = 3; i<11; ++i){
    if(i%2 == 0){
        jd[i] = jd[i-1]+3;
    }

    if(i%2 != 0){
        jd[i] = jd[i-1]+2;
    }
}

for(k = 1; k<(T/5-3)+1; ++k){
    t = i;
    jd[t] = jd[t-1] + 10;
    ++t;
    jd[t] = jd[t-1] + 11;
    ++t;
    for(i = t; i<t+8; ++i){
        if(i%2 == 0){
            jd[i] = jd[i-1]+13;
        }
        if(i%2 != 0){
            jd[i] = jd[i-1]+12;
        }
    }
}

printf("jd dizisi jdson.dat adli dosyaya yazdiriliyor...\n");
for(i=1; i<SN; ++i){
    fprintf(jdpp,"%d\n", jd[i]);
}

fclose(jdpp);

```

```

printf("JD dizisi jdson.dat adli dosyaya yazdirildi\n");
printf("jd dizisinin boyutu = %d\n",i-1);

}

void forceson(float *beta)
{

/* Kuvvet matrisinin hesaplanması ve fson.dat adlı dosyaya yazdırılması işleminin yapıldığı alt
fonksiyon*/

int i;
float f3, f4, ZER = 0.;
FILE *fpp;

fpp = fopen("fson.dat","w");
if(fpp == NULL){
printf("dosya acilis hatasi!");
exit(1);
}

printf("f dosyasi yazdiriliyor...\n");
f3 = *beta * -1;
f4 = *beta * -1;
for(i=1; i<T+1; ++i){
if(i%5 != 0){
fprintf(fpp, "%f\n", ZER);
fprintf(fpp, "%f\n", ZER);
}
else{
fprintf(fpp, "%f\n", ZER);
fprintf(fpp, "%f\n", f3 + f4);
}
}
fclose(fpp);
printf("f dosyasi Fson.dat adli dosyaya yazdirildi\n");
printf("f dosyasinin boyutu = %d\n",(i-1)*2);
}

```

ÖZGEÇMİŞ**Doğum Tarihi** 12.11.1974

Doğum Yeri Hollanda

Lise 1986 – 1993**Bolu Abant İzzet Baysal Anadolu Lisesi****Lisans** 1993 – 1997**Karadeniz Teknik Üniversitesi
Mühendislik – Mimarlık Fak.
Makina Mühendisliği Bölümü****Yüksek Lisans** 1997 – 2000**Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Makina Müh. Anabilim Dalı,
Konstrüksiyon Programı****Çalıştığı kurum(lar)**

1997 - 1999

Tanrıverdi Fermuar A.Ş.

1999 – Devam Ediyor

Bosal – Mimaysan A.Ş.

83
7
—
90