

83009

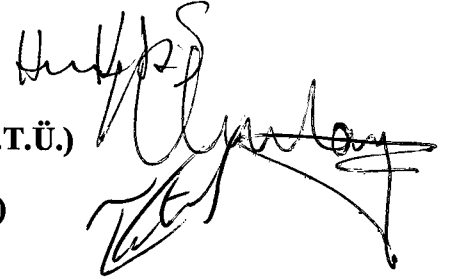
ANSYS PROGRAMI İLE DİZAYN OPTİMİZASYONU

YÜKSEK LİSANS TEZİ
Makina Müh. Engin AKHOROZ
(0396Y062)

83009

Tezin Enstitüye Verildiği Tarih : 6 Ocak 1999
Tezin Savunulduğu Tarih : 4 Şubat 1999

Tez Danışmanı : Doç.Dr. Hikmet KOCABAŞ
Diğer Jüri Üyeleri Prof.Dr. Teoman KURTAY (İ.T.Ü.)
Doç.Dr. Temel KOTİL (İ.T.Ü.)



TC. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

OCAK 1999

ÖNSÖZ

Teknolojideki hızlı gelişmelere paralel olarak kaynakları ekonomik kullanmak da önem kazanmıştır. Bu nedenle sistemleri dizayn ederken de bunları bir optimizasyon problemi gibi ele alıp çözüme kavuşturma arzusu kendini iyice hissettirmeye başlamıştır. Bu tezde ANSYS programı ile dizayn optimizasyonunun nasıl yapılacağı, işlem safhaları ve uygulamalarla anlatılmıştır.

Bu çalışma süresince değerli katkılarını esirgemeyen, danışman hocam sayın Doç. Dr. Hikmet Kocabaş'a teşekkürü borç bilirim.

Ocak 1999

Engin AKHOROZ

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	ii
İÇİNDEKİLER	iii
SEMBOL LİSTESİ	vii
ŞEKİL LİSTESİ	ix
TABLO LİSTESİ	xi
ÖZET	xii
SUMMARY	xiii
BÖLÜM 1: GİRİŞ	1
1.1 Dizayn Optimizasyonunun Gelişimi ve Sistematiği	1
1.2 Bilgisayarların Optimum Dizaynda Oynadığı Rol	2
BÖLÜM 2: DİZAYN OPTİMİZASYONU TEORİSİ	3
2.1 Optimum Dizayn Problemlerinin Formüle Edilmesi	3
2.1.1 Dizayn değişkenleri	4
2.1.2 Hedef fonksiyonu	5
2.1.3 Dizayn sınırlamaları	6
2.1.3.1 Doğrusal (linear) ve doğrusal olmayan (nonlinear) sınırlamalar	6
2.1.3.2 Eşitlik (equality) ve eşitsizlik (inequality) sınırlamaları	6
2.1.4 Sınırlanmamış (unconstrained) ve sınırlanmış (constrained) optimizasyon problemlerinin formüle edilmesi	8
2.2 Minimum	12
2.2.1 Genel (global) minimum	12
2.2.2 Yerel (local) minimum	12
2.2.3 Weierstrass teoremi	14
2.2.4 Eğim (gradient) vektörü	14
2.2.5 Hessian matrisi	15
2.3 Sınırlanmamış Fonksiyonların Yerel Minimumlarının Analitik Yolla Bulunması	15
2.3.1 Tek değişkenli sınırlanmamış bir fonksiyonun yerel minimumu için gerekli ve yeter durumlar	16
2.3.1.1 Birinci mertebeden gerekli durum	16

2.3.1.2 Yeter durumlar	16
2.3.1.3 İkinci mertebeden gerekli durum	16
2.3.2 Çok değişkenli sınırlanmamış bir fonksiyonun yerel minimumu için gerekli ve yeter durumlar	17
2.3.2.1 Gerekli durum	17
2.3.2.2 İkinci mertebeden gerekli durum	17
2.3.2.3 İkinci mertebe yeter durum	18
2.4 Sınırlanmış Fonksiyonların Yerel Minimumlarının Analitik Yolla Bulunması	18
2.4.1 Lagrange çarpanları (Lagrange multipliers)	18
2.4.2 Kuhn-Tucker gerekli durumları	19
2.4.3 Genel optimum	20
2.4.4 Fonksiyon ve setlerin konveksliği	21
2.4.5 Sınırlanmış fonksiyonlar için ikinci mertebe durumlar	22
2.4.5.1 Konveks problemler için yeter durumlar	22
2.4.5.2 Genel problemler için ikinci mertebe durumlar	23
2.4.5.2.1 Kuvvetli yeter durum	23
2.4.5.2.2 Yeter durumlar	23
2.5 Sınırlanmamış Problemlerin Optimum Dizaynı için Sayısal Yöntemler	23
2.5.1 Altın bölge metodu (Golden section method)	24
2.5.2 Dikey azalma metodu (Steepest descent method)	26
2.6 Sınırlanmış Problemlerin Optimum Dizaynı için Sayısal Yöntemler	28
2.6.1 Dolaylı çözüm metotları	28
2.6.1.1 Dış penaltı fonksiyonu metodu	28
2.6.1.2 İç penaltı fonksiyonu metodu	30
2.6.1.3 Artırmalı Lagrange çarpanları metodu	32
2.6.2 Direk çözüm metotları	32
BÖLÜM 3: ANSYS PROGRAMI İLE DİZAYN OPTİMİZASYONU	34
3.1 Giriş	34
3.2 ANSYS Terminolojisinde Optimizasyon için Kullanılan Temel Kavramlar	34
3.2.1 Dizayn değişkenleri (DVs : Design variables)	34
3.2.2 Durum değişkenleri (SVs :State variables)	35
3.2.3 Hedef fonksiyonu (Objective function)	35
3.2.4 Uygun ve uygun olmayan dizayn (Feasible & Infeasible design)	35
3.2.5 En iyi dizayn (Best design)	35
3.2.6 Analiz dosyası (Analysis file)	36
3.2.7 Döngü (Loop)	36
3.2.8 Optimizasyon iterasyonu (Optimization iteration)	36
3.2.9 Optimizasyon veritabanı (Optimization database)	36
3.3 Optimizasyon Değişkenlerinin Seçiminde Dikkat Edilecek Hususlar	37
3.3.1 Dizayn değişkenlerinin seçimi	37
3.3.2 Dizayn sınırlamalarının seçimi	39
3.3.3 Hedef fonksiyonunun seçimi	40
3.4 ANSYS ile Dizayn Optimizasyonunun Prosedürü	41
3.4.1 Adım 1- Analiz dosyasının oluşturulması	42
3.4.1.1 Modeli parametrik olarak oluşturmak	43

3.4.1.2 Çözümü yapmak	44
3.4.1.3 Sonuçların elde edilmesi	45
3.4.1.4 Analiz dosyasının hazırlanması	46
3.4.2 Adım 2- Optimizasyon için parametrelerin tanıtılması	47
3.4.3 Adım 3- Optimizasyon modülüne giriş ve bunda kullanılacak analiz dosyasının belirtilmesi	48
3.4.4 Adım 4- Optimizasyon değişkenlerinin belirtilmesi	49
3.4.5 Adım 5- Optimizasyon aracı veya metodunun seçilmesi	50
3.4.6 Adım 6- Optimizasyon döngü kontrollerinin belirtilmesi	50
3.4.7 Adım 7- Optimizasyon analizinin başlatılması	52
3.4.8 Adım 8- Dizayn setleri sonuçlarının incelenmesi	54
3.4.9 Dizayn setlerinin manipüle edilmesi	57
3.4.10 Optimizasyonun birden fazla sayıda çalıştırılması	58
3.4.10.1 Aynı oturumda birden fazla optimizasyonun yapılması	58
3.4.10.2 Farklı oturumlarda birden fazla optimizasyonun yapılması	58

BÖLÜM 4: ANSYS İLE OPTİMİZASYON UYGULAMALARI 61

4.1 Uygulama-1	61
4.1.1 Adım 1- İş adının (jobname) belirtilmesi	62
4.1.2 Adım 2- Analiz başlığının verilmesi	63
4.1.3 Adım 3- Parametrelere başlangıç değerlerinin atanması	63
4.1.4 Adım 4- Eleman tipinin tanımlanması	64
4.1.5 Adım 5- Gerçek sabitlerin (Real constants) tanımlanması	65
4.1.6 Adım 6- Malzeme özelliklerinin tanımlanması	66
4.1.7 Adım 7- Node 'ların oluşturulması	68
4.1.8 Adım 8- Elemanların oluşturulması	70
4.1.9 Adım 9- Yer değiştirme sınırlamaları ve yüklerin uygulanması	71
4.1.10 Adım 10- Modelin çözülmesi	75
4.1.11 Adım 11- Postprocessor 'e giriş ve toplam eleman hacminin elde edilmesi	75
4.1.12 Adım 12- Çubuklarda oluşan eksenel gerilmelerin elde edilmesi	77
4.1.13 Adım 13- Halihazırdaki dizaynın görüntülenmesi	79
4.1.14 Adım 14- Optimizasyon için analiz dosyasının oluşturulması	79
4.1.15 Adım 15- Optimizer 'a giriş ve analiz dosyasının tanıtılması	80
4.1.16 Adım 16- Optimizasyon dizayn değişkenlerinin tanımlanması	80
4.1.17 Adım 17- Sınırlamaların tanımlanması	82
4.1.18 Adım 18- Optimizasyon veritabanının saklanması	83
4.1.19 Adım 19- Ağırlığın hedef fonksiyonu olarak tanımlanması	86
4.1.20 Adım 20- Birinci mertebe optimizasyon metodunun belirtilmesi	85
4.1.21 Adım 21- Optimizasyonun çalıştırılması	86
4.1.22 Adım 22- En iyi dizayn seti ve tüm dizayn setlerinin listelenmesi	86
4.1.23 Adım 23- Ağırlığın (WT) ve taban uzunluğunun (B) iterasyon sayısının bir fonksiyonu olarak görüntülenmesi	88
4.1.24 Adım 24- Maksimum gerilme ve kesit alanının iterasyon sayısının bir fonksiyonu olarak görüntülenmesi	90
4.1.25 Adım 25- ANSYS 'den çıkış	93
4.2 Problemin Batch Dosyası Hazırlanarak Çözülmesi	93

4.3 Uygulama-2	98
4.4 Uygulama-3	105
BÖLÜM 5: ANSYS PROGRAMINDA KULLANILAN OPTİMİZASYON TEKNİKLERİ	113
5.1 Giriş	113
5.2 Optimizasyon Metotları	115
5.2.1 Altproblem yaklaşımı (Subproblem approximation) metodu	116
5.2.1.1 Fonksiyon yaklaşımları	117
5.2.1.2 Altproblem yaklaşımının minimum yapılması	118
5.2.1.3 Yaklaşım	122
5.2.1.4 Altproblem yaklaşımı metodu için bazı önemli hususlar	123
5.2.2 Birinci mertebe (First order) metodu	123
5.2.2.1 Sınırlanmamış hedef fonksiyonu	124
5.2.2.2 Arama yönü	125
5.2.2.3 Yaklaşım	127
5.2.2.4 Birinci mertebe metodu için bazı önemli hususlar	128
5.3 Optimizasyon Araçları	129
5.3.1 Tek döngü analiz aracı (Single loop analysis tool)	129
5.3.2 Rasgele dizayn aracı (Random tool)	129
5.3.3 Süpürme aracı (Sweep tool)	130
5.3.4 Faktöriyel aracı (Factorial tool)	130
5.3.5 Eğim aracı (Gradient tool)	131
BÖLÜM 6: SONUÇLAR ve ÖNERİLER	133
KAYNAKLAR	135
EK	136
ÖZGEÇMİŞ	137

SEMBOL LİSTESİ

d_o	: Dış çap
d_i	: İç çap
σ	: Gerilme
σ_a	: Emniyet gerilmesi
t, t_1, t_2	: Et kalınlığı
ρ	: Malzeme yoğunluğu
n	: Dizayn değişkeni sayısı
$x=(x_1, x_2, \dots, x_n)$	: Dizayn değişkenleri vektörü
x_{il}	: Dizayn değişkeni alt sınırı
x_{iu}	: Dizayn değişkeni üst sınırı
$f(x)$	: Hedef fonksiyonu
$h_i(x)$	: Eşitlik sınırlamaları
$g_i(x)$	: Eşitsizlik sınırlamaları
x^*	: Hedef fonksiyonunu minimum yapan dizayn değişkeni vektörü
$f(x^*)$	: Hedef fonksiyonunun minimum değeri
$f'(x)$	: Hedef fonksiyonunun birinci mertebe türevi
$f''(x)$	: Hedef fonksiyonunun ikinci mertebe türevi
S	: Uygun (feasible) bölge
$\nabla f(x)$	: x noktasındaki eğim vektörü
H	: Hessian matrisi ($f(x)$ 'in ikinci mertebe türevlerini içerir.)
λ	: Matrisin özdeğerleri
L	: Lagrange denklemi
$\sum$	: Toplam sembolü
v	: Eşitlik sınırlaması için Lagrange çarpanı
u	: Eşitsizlik sınırlaması için Lagrange çarpanı
$\nabla^2 L$	: Lagrange denkleminin ikinci mertebe türevi
∇h_i	: Eşitlik sınırlamalarının eğim vektörü
∇g_i	: Eşitsizlik sınırlamalarının eğim vektörü
A^T	: Herhangi bir A vektörü veya matrisinin transpozesi
d_q	: q . iterasyondaki yön vektörü
α	: İleri fark adım büyüklüğü
$P(x)$	: Penalte fonksiyonu
$A(x, \lambda, r_i)$	: Artırımlı Lagrange denklemi
λ	: Lagrange çarpanı
DVs	: Dizayn değişkenleri (Design variables)
SVs	: Dizayn sınırlamaları (State variables)
E	: Young modülü
M	: Moment
$\sigma_{\max}$	: Maksimum gerilme
$\delta_{\max}$	: Maksimum yerdeğiştirme (deformasyon)
ν	: Poisson oranı
$\alpha_i, \beta_i, \gamma_i$	: SVs toleransları

$\hat{f}(x)$	: Yaklaşık hedef fonksiyonu
$\hat{g}(x), \hat{h}(x), \hat{w}(x)$	: Yaklaşık sınırlama fonksiyonları
a_i, b_{ij}	: Yaklaşım fonksiyonu katsayıları
E^2	: Ağırlıklı en küçük kareler normu
$\phi^{(j)}$	: j. Dizayn seti ile ilgili ağırlık
n	: Dizayn seti sayısı
n_d	: Aktif dizayn seti sayısı
$F(x, p_k)$	: Penaltı fonksiyonları eklenmiş olan sınırlanmamış hedef fonksiyonu (Altproblem yaklaşımı metodu için)
$X(x_i)$	: Dizayn değişkeni sınırlamaları için penaltı fonksiyonu
$G(\hat{g}_i), H(\hat{h}_i), W(\hat{w}_i)$	: SVs sınırlamaları için penaltı fonksiyonu
c_1, c_2, c_3, c_4	: Penaltı fonksiyonu katsayıları
d_1, d_2, d_3, d_4	: Penaltı fonksiyonu katsayıları
ε	: Çok küçük olan bir pozitif sayı
$\tilde{F}^{(j)}$	: j. iterasyondaki sınırlanmamış minimum hedef fonksiyonu
$\tilde{x}^{(j)}$	: $\tilde{F}^{(j)}$ 'ye karşılık gelen dizayn değişkeni vektörü
τ, ρ_i	: Hedef fonksiyonu ve dizayn değişkeni toleransları
n_s, n_i	: Altproblem iterasyonlarının sayısı
n_{si}	: Art arda oluşan dizayn setlerinin sayısı.
N_s, N_r	: Maksimum iterasyon sayısı.
N_{si}	: Art arda oluşan, maksimum uygun olmayan dizayn setlerinin sayısı
$Q(x, q)$	: Penaltı fonksiyonları eklenmiş olan sınırlanmamış hedef fonksiyonu (Birinci mertebe metodu için)
P_x	: Dizayn değişkenlerine uygulanan dış penaltı fonksiyonu
P_g, P_h, P_w	: SVs sınırlamaları için genişletilmiş iç penaltı fonksiyonları
λ	: Çok büyük bir tamsayı
s_j	: Çizgi arama parametresi (adım büyüklüğü)
N_l	: Müsaade edilen maksimum iterasyon sayısı
n_r	: Herbir çalıştırmada oluşturulan rasgele iterasyonları sayısı
n_f	: Toplam uygun dizayn sayısı (Rasgele dizayn aracı için)
N_f	: Arzulanan uygun dizayn seti sayısı (Rasgele dizayn aracı için)
ΔD	: İleri fark adım büyüklüğü

ŞEKİL LİSTESİ

		Sayfa No
Şekil 1.1	Optimum dizayn işlemi	2
Şekil 2.1	Çeşitli kesitler ve bunlara ait dizayn parametreleri	4
Şekil 2.2	(a) Minimum bulma ($f(x)$), (b) maksimum bulma ($-f(x)$)	6
Şekil 2.3	Eşitlik ve eşitsizlik sınırlamaları arasındaki fark	7
Şekil 2.4	İki çubuklu destek sistemi	9
Şekil 2.5	Optimum noktaların grafiksel gösterimi. (a) Sınırsız fonksiyon, (b)Sınırlı fonksiyon	13
Şekil 2.6	Bir minimumun kesin olarak bulunduğu aralığın tespiti	25
Şekil 2.7	Belirlenen aralığın üç eşit parçaya ayrılması	25
Şekil 2.8	$F(x)=0.5.x$ fonksiyonunun $4 - x \leq 0$ sınırlamasına göre dış penaltı fonksiyonu	29
Şekil 2.9	$F(x)=0.5.x$ fonksiyonunun $4 - x \leq 0$ sınırlamasına göre iç penaltı fonksiyonu	30
Şekil 2.10	Genişletilmiş iç penaltı fonksiyonu	31
Şekil 3.1	Optimizasyonda bilgi akışı	37
Şekil 3.2	Bir ankastre giriş için dizayn değişkenlerinin seçilmesi	38
Şekil 3.3	Dizayn optimizasyonu için bir örnek	42
Şekil 4.1	Problemin şematik gösterilişi	62
Şekil 4.2	İş adı diyalog kutusu	62
Şekil 4.3	Analiz başlığı diyalog kutusu	63
Şekil 4.4	Parametrelere başlangıç değerlerinin atanması	63
Şekil 4.5	Eleman tipleri diyalog kutusu	64
Şekil 4.6	Eleman tipleri kütüphanesi	65
Şekil 4.7	Gerçek sabitler diyalog kutusu	65
Şekil 4.8	Gerçek sabitler için eleman tipi diyalog kutusu	66
Şekil 4.9	LINK1 elemanı için gerçek sabitler diyalog kutusu	66
Şekil 4.10	İzotropik malzeme özellikleri diyalog kutusu	67
Şekil 4.11	Malzeme özelliklerinin belirtilmesi	67
Şekil 4.12	Aktif koordinat sisteminde node oluşturma	68
Şekil 4.13	Çizilen nodların grafik penceresindeki görüntüsü	69
Şekil 4.14	Elemanların özelliklerinin belirtildiği diyalog kutusu	70
Şekil 4.15	Çizilen elemanların grafik penceresindeki görüntüsü	71
Şekil 4.16	Nodları sınırlama diyalog kutusu	72
Şekil 4.17	Sınırlamaların tipinin belirtildiği diyalog kutusu	72
Şekil 4.18	Nodlara kuvvet uygulama diyalog kutusu	73
Şekil 4.19	Kuvvetlerin yön ve büyüklüklerinin belirtildiği diyalog kutusu	73
Şekil 4.20	Sınırlama ve kuvvetlerin grafik ekrandaki görüntüleri	74
Şekil 4.21	Eleman tablosu bilgisi diyalog kutusu	75
Şekil 4.22	Eleman tablosu oluşturulacak itemin belirtilmesi	76
Şekil 4.23	Skalar veri elde etme diyalog kutusu	76
Şekil 4.24	Eleman tablosu toplamı sonuçlarının elde edilmesi	77
Şekil 4.25	Şekil ve büyüklüğün düzenlenebildiği diyalog kutusu	79
Şekil 4.26	Analiz dosyasının belirtilmesi	80
Şekil 4.27	Dizayn değişkenleri diyalog kutusu	81
Şekil 4.28	Dizayn değişkenlerinin tanımlandığı diyalog kutusu	81

Şekil 4.29	SVs tanımlanması	82
Şekil 4.30	Tanımlanan SVs	83
Şekil 4.31	Oluşturulan optimizasyon bilgilerinin saklanması	84
Şekil 4.32	Hedef fonksiyonunun tanımlanması	84
Şekil 4.33	Optimizasyon metodunun tanımlanması	85
Şekil 4.34	Birinci merteye metodu kontrolleri	85
Şekil 4.35	Grafik kontrolleri diyalog kutusu	89
Şekil 4.36	Grafikte yer alacak parametrelerin belirtilmesi	89
Şekil 4.37	İterasyon sayısına bağlı olarak sistem ağırlığında meydana gelen değişim	90
Şekil 4.38	İterasyon sayısına bağlı olarak taban uzunluğunda meydana gelen değişim	91
Şekil 4.39	İterasyon sayısına bağlı olarak elemanlarda oluşan maksimum gerilmede meydana gelen değişim	92
Şekil 4.40	İterasyon sayısına bağlı olarak elemanların kesit alanlarında meydana gelen değişim	93
Şekil 4.41	Optimizasyonu yapılacak giriş	98
Şekil 4.42	Optimizasyonu yapılacak ankastre giriş	105
Şekil 4.43	Anahtar noktaları (keypoints), alanları, ve TK16 isimli dizayn değişkenini gösteren yarı simetrik katı model	106
Şekil 4.44	Toplam hacmin iterasyon sayısı ile değişimi (altproblem yaklaşımı metodu)	110
Şekil 4.45	Toplam hacmin iterasyon sayısı ile değişimi (birinci merteye metodu)	110
Şekil 4.46	Kirişte oluşan deformasyonun iterasyon sayısı ile değişimi (altproblem yaklaşımı metodu)	111
Şekil 4.47	Kirişte oluşan deformasyonun iterasyon sayısı ile değişimi (birinci merteye metodu)	111
Şekil 4.48	Maksimum gerilmenin iterasyon sayısı ile değişimi (altproblem yaklaşımı metodu)	112
Şekil 4.49	Maksimum gerilmenin iterasyon sayısı ile değişimi (birinci merteye metodu)	112
Şekil 5.1	SVs uzayında genişletilmiş iç penaltı fonksiyonu	121

TABLO LİSTESİ

		<u>Sayfa No</u>
Tablo 4.1	En iyi dizayn seti	86
Tablo 4.2	Uygulama-1 için oluşan bütün dizayn setleri	87
Tablo 4.3	Uygulama-2 için oluşan bütün dizayn setleri	104
Tablo A.1	ANSYS Optimizasyon modülünde kullanılan komutlar	135



ANSYS PROGRAMI İLE DİZAYN OPTİMİZASYONU

ÖZET

Mühendislik bilimi, analiz, dizayn, fabrikasyon, satış, araştırma ve geliştirme gibi birçok faaliyet alanı içermektedir. Bunlardan biri olan “sistem dizaynı”, mühendislikte büyük bir yer kaplamaktadır. Hızla gelişmekte olan dünyada, sadece çalışan bir sistem geliştirmek artık tatmin edici olmaktan uzaklaşmaktadır. Önemli olan “*en iyi*” sistemi geliştirmektir. “En iyi” kavramına, gördüğü işlevlere göre en hafif, en ucuza mal olmuş en verimli, en hızlı, çok fonksiyonlu, dayanıklı vb. manalar atfedilebilir. Böyle bir sistemin dizaynı, “optimizasyon problemi” olarak formüle edilip çözümlenebilir. En basit tabiri ile bir optimizasyon problemi, “en iyi” den kastedilen manaları hedef olarak alıp, matematiksel fonksiyonla temsil ettirilerek, belirtilen sınırlar dahilinde bu fonksiyonun minimum (bazen maksimum) değerinin bulunmasıdır. Böyle bir problemin üç önemli bileşeni vardır:

- *Dizayn değişkenleri:* Genellikle uzunluk, kalınlık, çap vb. gibi modeli tanımlayan geometrik büyüklüklerdir. Bunlar bağımsız değişkenlerdir.
- *Dizayn sınırlamaları:* Dizayn değişkenlerine bağımlı olarak tanımlanan ve sisteme ait gerilme, frekans, boyut, deformasyon, sıcaklık vb. büyüklüklerdir sınırlarını belirten değişkenlerdir.
- *Hedef fonksiyonu:* Dizayn değişkenlerine bağımlı olarak tanımlanan ve dizayn sınırlamaları dahilinde minimum veya maksimum yapılmaya çalışılan fonksiyondur.

Optimizasyon problemlerin çözümünde analitik ve sayısal yöntemler kullanılmaktadır. Analitik yöntemlerin, çözüm yolunda belirli bir aşamadan daha ileriye geçememesi birçok sayısal yöntemin geliştirilmesine zemin hazırlamıştır. Sayısal yöntemlerde de tekrarlı (iteratif) işlemlerin çokça yapılması, çağımızın en önde gelen vazgeçilmezlerinden olan bilgisayarları devreye sokmuştur. CAD, CAM, CAE programları üreten bazı büyük yazılım şirketleri, geliştirdikleri bu programlara dizayn optimizasyonu yapan modüller de eklemişlerdir. Bu tezde dizayn optimizasyonunun teorisi ve ANSYS programı ile nasıl yapılacağı ele alınmıştır.

DESIGN OPTIMIZATION USING ANSYS

SUMMARY

The goal of many engineers is to design systems for automotive, aerospace, mechanical, civil, chemical, industrial, electrical, biomedical, agricultural, naval and nuclear engineering applications. In the highly competitive world today, it is no longer sufficient to design a system that performs the required task satisfactorily. It is essential to design the *best system*. “Best” means an efficient, versatile, unique and cost effective system. To design such systems, proper analytical, experimental and numerical methods are needed. Optimum design concepts and methods provide some of the needed tool.

The design of systems can be formulated as problems of optimization where a measure of performance is to be optimized while satisfying all the constraints. In recent years, numerical methods of optimization have been developed extensively. Many of the methods have been used to design better systems.

Any problem in which certain parameters need to be determined to satisfy constraints can be formulated as an optimum design problem. Once this has been done, optimization methods can be used to solve the problem. Figure 1 shows the optimum design process.

In an optimization problem, there are three basic terms:

1. Design variables
2. Objective function
3. Design constraints (“State variables” in ANSYS terminology)

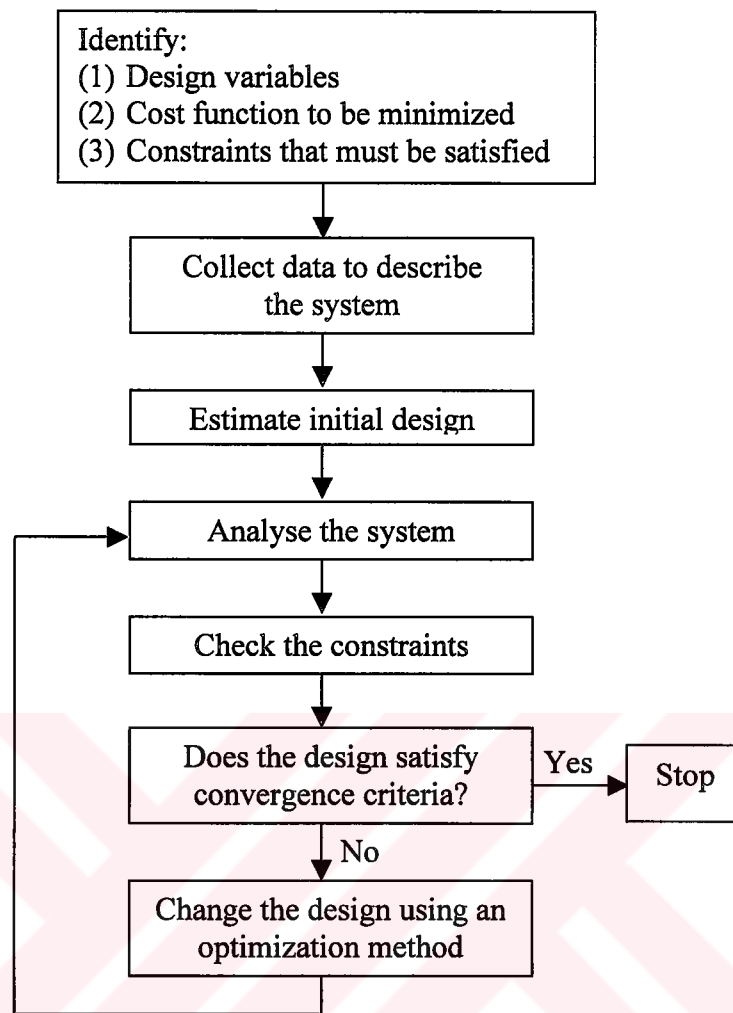


Figure 1- Optimum design process

Design variables: Parameters chosen to describe the design of a system are called the design variables. These variables are independent, so designer can assign any value to them. If the specified values do not satisfy all constraints, the design is *not feasible*, which is called an *infeasible design*. If the constraints are satisfied, we have a *feasible (workable or usable) design*. All design variables for a problem are represented in the vector x .

Objective function: There can be many feasible designs for a system and some are better than others. To make such a claim we must have some criterion to compare various designs. The criterion must be a scalar function whose numerical value can be obtained once a design is specified i.e. it must be a function of the design variables. The objective function is represented by f , or $f(x)$ to emphasize its

dependence on design variables. Numerical methods was developed to minimize the objective function. If the function is desired to be maximum, it is converted from $f(x)$ to $-f(x)$. Then $-f(x)$ is minimized by the same methods.

Design constraints: All restrictions placed on a design are collectively called constraints. Each constraints must be influenced by one or more design variables. Some constraints are quite simple, such as minimum and maximum of design variables, while more complex ones may be indirectly influenced by design variables. Design problem may have equality as well as inequality constraints. For example, to perform the desired operation, a machine component must move precisely by Δ , so we must treat this as an equality constraint. A feasible design must satisfy precisely all the equality constraints. Also, there are inequality constraints in most design problems. Examples of such constraints are that calculated stresses must not exceed allowable stress of the material, deflections must not exceed specified limits, resources must not be exceeded, etc.

The standart design optimization model is defined as follows: Find an n-vector $x=\{x_1, x_2, \dots, x_n\}$ of design variables to minimize a cost function

$$f(x)=f(x_1, x_2, \dots, x_n)$$

subject to p equality constraints

$$h_j(x) \equiv h_j(x_1, x_2, \dots, x_n) = 0 \quad j=1 \text{ to } p$$

and m inequality constraints

$$g_i(x) \equiv g_i(x_1, x_2, \dots, x_n) \leq 0 \quad i=1 \text{ to } m$$

Analytical methods for solving some problems are not applicable at all. For that reason many numerical methods are developed.

The design of complex systems requires large calculations and data processing. Today's computers can perform complex calculations and process large amounts of

data efficiently. Better systems can now be designed by analysing various options in a short time.

When optimization problems are properly implemented into a software, they give powerful numerical tools for designing best systems.

This thesis focuses on optimization module of ANSYS software program. Furthermore, some theory of analytical and numerical methods are mentioned to understand the sense of optimization better.

The ANSYS program offers two optimization methods to accomodate a wide range of optimization problems. The “*subproblem approximation method*” is an advanced zero-order method that can be efficiently applied to most engineering problems. The “*first order method*” is based on design sensitivities and is more suitable for problems that require high accuracy.

For both the subproblem approximation and first order methods, the program performs a series of analysis-evaluation-modification cycles. That is, an analysis of the initial design is performed, the results are evaluated against specified design criteria, and the design is modified as necessary. This process is repeated until all specified criteria are met.

In addition to the two optimization techniques available, the ANSYS program offers a set of strategic tools that can be used to enhance the efficiency of the design process. For example, a number of random design iterations can be performed. The initial data points from the random design calculations can serve as starting points to feed the optimization methods mentioned above.

There are basically two ways to approach an ANSYS optimization: as a batch run or interactively through the Graphical User Interface (GUI).

The usual procedure for design optimization consists of the following main steps. These steps may vary slightly, depending on whether you are performing optimization interactively (through the GUI) or in batch mode.

1. Create an analysis file to be used during looping. This file should represent a complete analysis sequence and must do the following:
 - a. Build the model parametrically (in PREP7 module)
 - b. Obtain the solution(s) (SOLUTION)
 - c. Retrieve and assign to parameters the response quantities that will be used as state variables (design constraints) and objective functions (POST1/POST26)
2. Establish parameters in the ANSYS database which correspond to those used in the analysis file; this step is typical but not required (Begin or OPT)
3. Enter OPT and specify the analysis file (OPT)
4. Declare optimization variables (OPT)
5. Choose optimization tool or method (OPT)
6. Specify optimization looping controls (OPT)
7. Initiate optimization analysis (OPT)
8. Review the resulting design sets data (OPT) and postprocess results (POST1/POST26)

The optimization module (/OPT) is an integrated part of the ANSYS program that can be employed to determine the optimum design.

Design configurations that satisfy all constraints are referred to as *feasible designs*.

Design configurations with one or more violations are termed *infeasible*.

As design sets are generated by methods or tools and if an objective function is defined, the *best design* set is computed and its number is stored.

The subproblem approximation method can be described as an advanced zero-order method in that it requires only the values of the dependent variables, and not their derivatives. There are two concepts which play a key role in the subproblem approximation method: the use of approximations for the objective function and state variables, and the conversion of the constrained optimization problem to an

unconstrained problem. The conversion is done by adding penalties to the objective function approximation to account for the imposed constraints. Because approximations are used for the objective function and SVs, the optimum design will be only as good as approximations.

Like the subproblem approximation method, the first order method converts the problem to an unconstrained one by adding penalty functions to the objective function.

The first order method uses gradients of the independent variables with respect to the design variables. For each iteration, gradient calculations are performed in order to determine a search direction, and a line search strategy is adopted to minimize the unconstrained problem.



BÖLÜM 1

GİRİŞ

1.1 Dizayn Optimizasyonunun Gelişimi ve Sistematiği

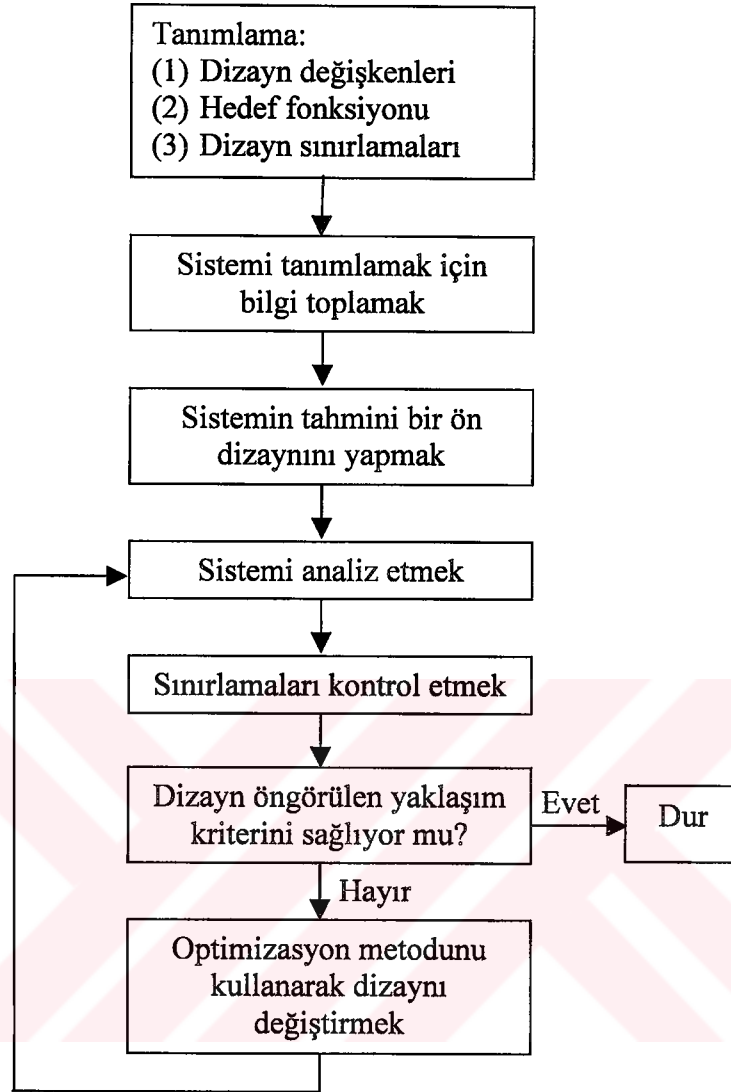
Mühendislik bilimi, analiz, dizayn, fabrikasyon, satış, araştırma ve geliştirme gibi birçok faaliyet alanı içermektedir. Bunlardan biri olan “sistem dizaynı”, mühendislikte büyük bir yer kaplamaktadır. Karmaşık sistemlerin dizaynı, yapılması uzun zaman alan birçok hesaplama gerektirmektedir.

Sistemlerin dizaynı “optimizasyon problemi” olarak formüle edilebilir. Böyle bir problemde dizayna ait tüm sınırlamalar sağlanmak şartı ile, bir değerin optimizasyonu sağlanır. Son yıllarda optimizasyonla ilgili bir çok sayısal metot geliştirilmiştir. İçerisinde sınırlamaları oluşturacak kesin parametreler bulunan herhangi bir problem, bir optimum dizayn problemi olarak formüle edilebilir. Yani bu teknikler geneldir ve çok farklı alanlarda kullanılabilir. Bir optimum dizayn Şekil 1.1 ‘deki işlem sırasına göre yapılır. [1]

1.2 Bilgisayarların Optimum Dizaynda Oynadığı Rol

Özellikle son 30 yılda, bilgisayar teknolojisi ve sayısal hesaplamalarda adeta bir devrim yaşandı. Günümüz bilgisayarları çok karmaşık hesaplamaları dahi, kısa süreler içerisinde ve yüksek doğruluk dereceleriyle çözebilmektedirler.

Mühendislikteki dizayn işlemi de bu devrimden ileri derecede yararlandı. Bu da arzulanan sistemi daha kısa sürede oluşturmak, kendisinden istenen kabiliyetleri de sağlayarak ucuza mal etmek vb. daha birçok avantajı beraberinde getirdi.



Şekil 1.1- Optimum dizayn işlemi. [1]

Mühendislik sistemleri, bilgisayarlar kullanılarak daha kesin olarak analiz edilebilir. Optimizasyon işleminde bulunan iterasyonlar nedeniyle çok büyük miktarlarda veriler meydana gelebilmektedir. Bu veriler, grafikler kullanılarak anlaşılabilir bir hale getirilebilir. Ayrıca renkli animasyonlar oluşturularak işlemin seyri de gözlenebilir. Örneğin; kuvvet basınç vb. etkiler altındaki bir cismin gerilme dağılımı, gerilmenin farklı bölgelerdeki büyüklüğüne göre değişik renkler kullanılarak elde edilebilir. Araç dinamiğinde bir aracın hareketini, araç imal edilmeden önce bilgisayar ekranında canlandırma (simulation) ile gözlemlemek mümkündür. [1]

BÖLÜM 2

DİZAYN OPTİMİZASYONU TEORİSİ

2.1 Optimum Dizayn Problemlerinin Formüle Edilmesi

Genel kabul gören bir ifadeye göre, bir problemi formüle etmek o problemi çözmek için gereken toplam çalışmaların %50 'sini oluşturur. Dizayn optimizasyonunda da problemin doğru formüle edilmesi çok önemlidir. Bir dizayn optimizasyonu problemini tanımlamak için üç şey gerekmektedir.

1. Dizayn değişkenleri
2. Dizayn sınırlamaları
3. Hedef fonksiyonu

İlk olarak sistemi tanımlamak için, dizayn değişkenleri olarak isimlendirilen bir değişken seti belirlenir. Daha sonra bu değişkenlere tahmini olarak sayısal değerler atanarak sistem oluşturulur. Bu aşamada sistemin arzu edilen biçimde çalışıp çalışmayacağı önemli değildir.

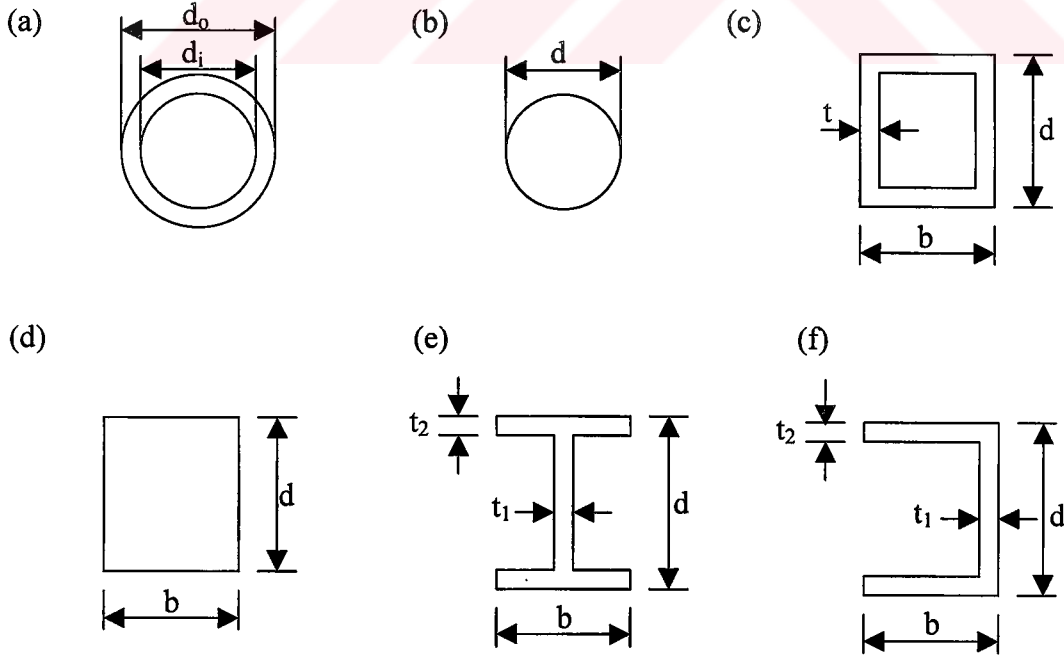
Bütün sistemler, kaynaklar, malzeme dayanımı, sistemin cevabı, sistem elemanlarının boyutları vb. bazı dizayn sınırlamaları içerisinde çalışacak şekilde tasarlanır. Bu sınırlamalar dizayn değişkenlerine bağımlı olarak tanımlanırlar. Eğer bir dizayn tüm sınırlamaları sağlıyorsa, sistem *uygun veya çalışabilir (feasible)* dir. Eğer belirtilen sınırlamalar sağlanamıyorsa sistem uygun değildir ve *uygun olmayan veya çalışmaz (infeasible)* olarak isimlendirilir. Bir uygun dizayn kullanışlıdır fakat başka uygun dizaynlar da olabileceğinden en iyisi olmayabilir.

Bir dizaynın, diğer bir dizayndan daha iyi olup olmadığını anlayabilmek için bir kriter gereklidir. Bu kriter *hedef fonksiyonu* denir. Hedef fonksiyonu da dizayn değişkenlerine bağımlı olarak tanımlanır.

Optimum çözümü elde edebilmek için doğru formülasyonu oluşturmak çok önemlidir. Mesela; kritik bir sınırlama unutulursa, bunun neticesinde büyük hataların ortaya çıkması kaçınılmazdır. Bazen de çok fazla sınırlama olması ve bunların birbirleri ile çelişmesi neticesinde, dizayn probleminin çözümü olmaz. [1]

2.1.1 Dizayn değişkenleri

Bir sistemin dizaynını tanımlamak için seçilen değişkenlere “dizayn değişkenleri” denir. Bu değişkenlere ilk olarak rasgele değerler verilerek sistem oluşturulur. Dizayn değişkenlerinin doğru seçilmesi gerekmektedir. Bunlar, mümkün olduğu kadar birbirlerinden bağımsız olmalıdırlar. Bazen bağımlı dizayn değişkenleri seçilmişse de, bu problem formülasyonunun gereksiz yere karmaşık hale gelmesine neden olur. Örneğin; Şekil 2.1.(a) ‘da ki boruyu ele alalım. [1]



Şekil 2.1- Çeşitli kesitler ve bunlara ait dizayn parametreleri [1]

Bir boruya ilk bakışta dizayn değişkenleri olarak iç çap (d_i), dış çap (d_o), ve et kalınlığı (t) söylenebilir. Fakat burada t 'nin d_i ve d_o 'ya bağımlı olduğu gözden kaçmamalıdır.

Mesela; $d_i=10$, $d_o=12$ ve $t=2$ olarak alınamaz. Çünkü $t=0.5(d_o-d_i)$ dir. Eğer problemi d_i , d_o ve t 'yi dizayn değişkenleri olarak alıp tanımlarsak, $t=0.5(d_o-d_i)$ 'yi de sınırlama olarak almak zorunda kalırız. Böylece problemimizi gereksiz yere karmaşık hale getirip zaman kaybetmiş oluruz. Dizayn değişkenleri olarak d_i ve d_o 'yu almak daha uygun ve yeterlidir. Şekil 2.1 'de bazı kesitler ve bunlara ait dizayn değişkenleri de gösterilmiştir.

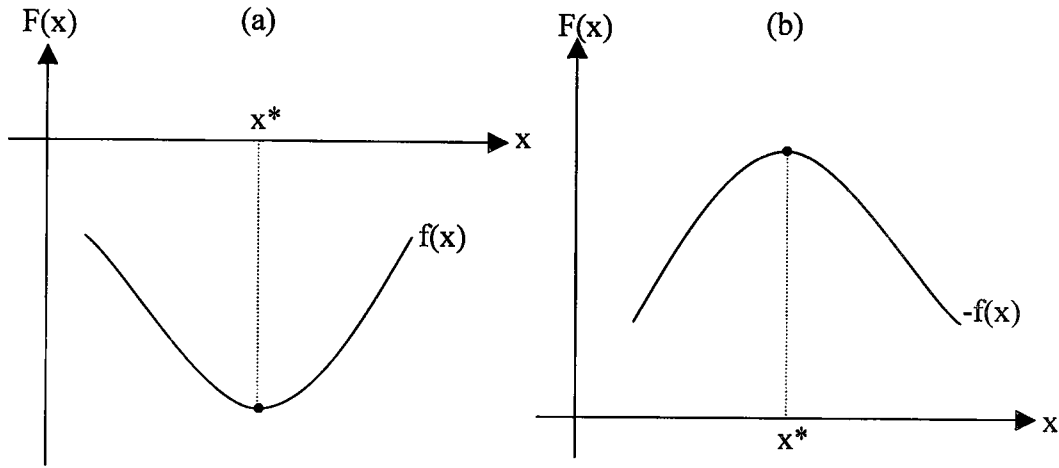
Özet olarak, bir problemin dizayn değişkenlerini belirlemek için aşağıdaki hususlar dikkate alınır:

1. Dizayn değişkenleri mümkün olduğu kadar birbirinden bağımsız olarak alınmalıdır.
2. Bir dizayn problemini doğru olarak formüle edebilmek için minimum bir sayıda dizayn değişkeni gerekir. [1]

2.1.2 Hedef fonksiyonu

Bir sistem ile ilgili birden fazla ve bazıları diğerlerinden daha iyi uygun dizayn olabilir. Bunları birbirleri ile kıyaslamak için bir kritere ihtiyacımız vardır. İşte bu kritere optimum dizayn problemi için "*hedef fonksiyonu*" denir. Bu fonksiyon dizayn değişkenlerine bağımlıdır. x dizayn değişkeni vektörü olmak üzere, hedef fonksiyonu $f(x)$ ile gösterilir.

Hedef fonksiyonunu doğru seçmek dizayn işleminde çok önemlidir. Minimum maliyet, maksimum kar, minimum ağırlık, minimum enerji gibi birçok hedef fonksiyonu kullanılmaktadır. Eğer hedef fonksiyonu maksimum kar problemindeki gibi bir maksimum bulma problemi ise $-f(x)$ olarak alınır (Şekil 2.2). [1]



Şekil 2.2- (a) Minimum bulma ($f(x)$), (b) maksimum bulma ($-f(x)$) [1]

Birçok dizayn problemlerinde, hedef fonksiyonu ve bunun dizayn değişkenleri ile nasıl bir ilişkisi olacağı açık değildir. Genelde bu tip problemler birçok alt probleme ayrılarak, her birinin dizayn optimizasyonu yapılarak çözülür. Örneğin bir binek arabanın optimizasyonu için dizayn değişkenleri nelerdir?, hedef fonksiyonu nedir? ve dizayn değişkenleri ile ifade edilecek fonksiyonel hali nasıldır? Bu oldukça karmaşık bir problemidir. Fakat problem, kapılar, yan paneller, tavan, süspansiyon sistemi, iletim sistemi vb. alt problemlere ayrılıp her biri optimize edilerek çözülebilir.

2.1.3 Dizayn sınırlamaları (constraints)

2.1.3.1 Doğrusal (linear) ve doğrusal olmayan (nonlinear) sınırlamalar

Birçok sınırlama fonksiyonu dizayn değişkenlerinin sadece birinci derecedeki terimlerini içerir. Bunlar doğrusal sınırlamalar olarak adlandırılır. Eğer dizayn değişkenlerinin daha yüksek derecedeki terimleri varsa, bunlara doğrusal olmayan sınırlamalar denir.

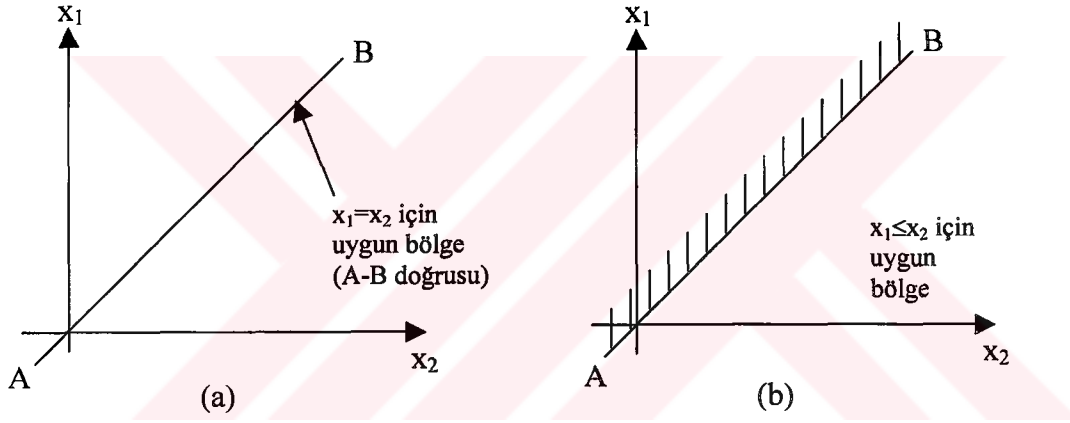
2.1.3.2 Eşitlik (equality) ve eşitsizlik (inequality) sınırlamaları

Dizayn problemleri eşitlik ve eşitsizlik sınırlamalarını içerebilirler. Örneğin; istenen operasyonu yapabilmek için bir makine parçasının tam olarak Δ kadar hareket etmesi

gerekliyorsa, bunu bir eşitlik sınırlaması olarak ele almamız gerekir. Uygun bir dizayn tüm eşitlik sınırlamalarını tam olarak sağlamalıdır.

Eşitsizlik sınırlamalarında ise, hesaplanan değerlerin bazı değerlerin üstüne çıkmaması veya bazı değerlerin altına düşmemesi istenir. Örneğin; gerilme değeri malzeme emniyet gerilmesi sınırını aşmamalı, malzemede oluşacak deformasyon belirtilen sınırları aşmamalı vb...

Eşitsizlik sınırlamalarına göre birçok uygun dizayn olabilir. Fakat eşitlik sınırlamalarına göre uygun dizayn bulmak daha zordur. [1]



Şekil 2.3- Eşitlik ve eşitsizlik sınırlamaları arasındaki fark [1]

Bu durumu daha iyi anlayabilmek için Şekil 2.3 ' ü inceleyelim. Şekil 2.3.(a) $x_1 = x_2$ eşitlik sınırlamasını göstermektedir. Bu sınırlamaya göre uygun dizaynlar A-B doğrusu üzerinde bulunmalıdır. Fakat sınırlama Şekil 2.3.(b)'de ki gibi $x_1 \leq x_2$ eşitsizlik sınırlaması şeklinde yazılırsa uygun bölge çok daha geniş olacaktır. A-B doğrusu üzerindeki veya altında kalan bölgenin içindeki herhangi bir nokta uygun dizaynı vermektedir.

2.1.4 Sınırlanmamış (unconstrained) ve sınırlanmış (constrained) optimizasyon problemlerinin formüle edilmesi

Optimizasyon problemleri sınırlanmamış (unconstrained) ve sınırlanmış (constrained) olmak üzere iki ana başlık altında toplanabilir. Adından da anlaşılacağı gibi sınırlanmamış optimizasyon, hedef fonksiyonu üzerinde hiçbir sınırlamanın olmadığı problemlerin optimum çözümünün bulunmasıdır. Sınırlanmış optimizasyon ise en az bir adet sınırlamanın tanımlandığı problemlerdir. Her iki problem tipi için de geliştirilmiş çözüm teknikleri vardır. İlk bakışta mutlaka sınırlamalar olacağı düşünülerek, sınırlanmamış problemlerin mühendislik sahasındaki bir dizayn problemi ile ilişkisi olmayacağı düşünülebilir. Fakat sınırlanmış problemler bazı tekniklerle, daha basit çözüm teknikleri içeren sınırlanmamış problemlere dönüştürülerek de çözülebilir.

Dizayn optimizasyonu problemlerinin genel tanımı aşağıdaki gibi yapılmaktadır.

$x = (x_1, x_2, \dots, x_n)^T$ dizayn değişkenleri vektörü olmak üzere ;

$$\triangleright f(x) = f(x_1, x_2, \dots, x_n) \quad \text{hedef fonksiyonunu} \quad (2.1)$$

$$\bullet h_i(x) = 0 \quad i = 1 \rightarrow p \quad \text{eşitlik sınırlamaları ve} \quad (2.2)$$

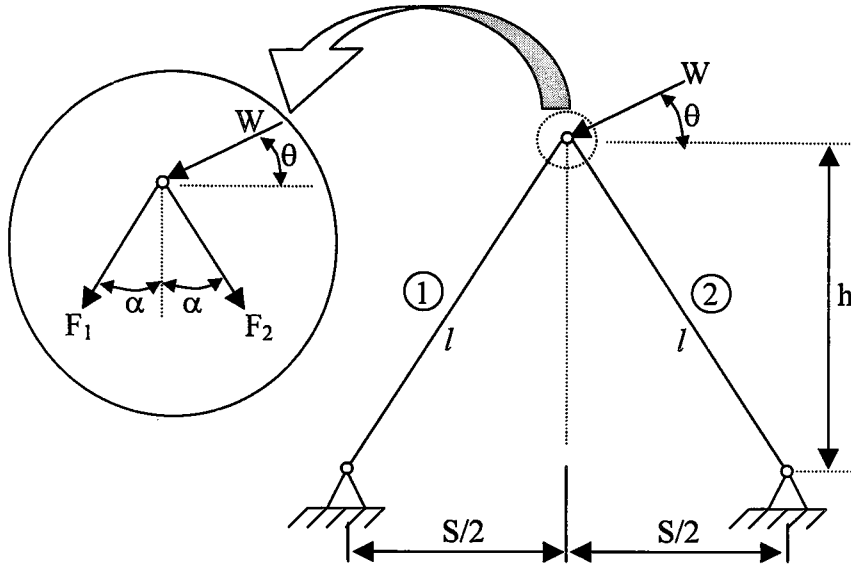
$$\bullet g_i(x) \leq 0 \quad i = 1 \rightarrow m \quad \text{eşitsizlik sınırlamalarına göre} \quad (2.3)$$

minimum yap.

Eğer sınırlanmamış bir optimizasyon problemi ise sadece;

$$\triangleright f(x) = f(x_1, x_2, \dots, x_n) \quad \text{hedef fonksiyonunu minimum yap.}$$

Şimdi yukarıda tanımlarını yaptığımız dizayn değişkenleri, hedef fonksiyonu ve dizayn sınırlamalarını daha iyi anlayabilmek için, ele alacağımız bir örnekle bunları oluşturalım. [1]



Şekil 2.4- İki çubuklu destek sistemi [1]

Yukarıdaki şekilde görülen iki çubuklu destek sistemine yapısal bir deformasyon olmamak kaydı ile 0° ile 90° arasındaki bir θ açısı ile W kuvveti etkiliyor olsun. Destek sisteminin yüksekliği h , taban genişliği ise S olsun. Bu desteklerden çok sayıda üretileceğinden dizaynın hedefi, gerekli sınırlamaları sağlayarak kütleyi, dolayısıyla harcanacak malzemeyi minimize etmektir.

F_1 ve F_2 kuvvetleri, deformasyon olmaksızın sistemin taşıyabileceği en büyük kuvvetler olsun. Statik denge denklemlerini kullanarak;

$$-F_1 \sin \alpha + F_2 \sin \alpha = W \cos \theta \quad (2.4)$$

$$-F_1 \cos \alpha - F_2 \cos \alpha = W \sin \theta \quad (2.5)$$

yazabiliriz. Sistemin geometrisinden hareketle $\sin \alpha = S / 2l$ ve $\cos \alpha = h / l$ olup denklemler aşağıdaki forma getirilebilir.

$$F_1 = -0.5Wl \left[\frac{\sin \theta}{h} + \frac{2 \cos \theta}{S} \right] \quad (2.6)$$

$$F_2 = -0.5Wl \left[\frac{\sin \theta}{h} - \frac{2 \cos \theta}{S} \right] \quad (2.7)$$

formüllerde yer alan l ise aşağıdaki gibi yazılabilir:

$$l = \sqrt{h^2 + (0.5S)^2}$$

❖ Dizayn değişkenleri:

- x_1 = destek sisteminin yüksekliği (h)
- x_2 = destek sisteminin taban genişliği (S)
- x_3 = 1. çubuğun dış çapı
- x_4 = 1. çubuğun iç çapı
- x_5 = 2. çubuğun dış çapı
- x_6 = 2. çubuğun iç çapı

olmak üzere dizayn değişkenleri vektörü:

$$x = (x_1, x_2, x_3, x_4, x_5, x_6)^T$$

❖ Hedef fonksiyonu (Kütle):

ρ kullanılan malzemenin yoğunluğu olmak üzere:

$$\text{➤ } f(x) = \frac{\pi \rho}{8} \sqrt{4 \cdot x_1^2 + x_2^2} \cdot (x_3^2 + x_5^2 - x_4^2 - x_6^2) \quad (2.8)$$

❖ Dizayn sınırlamaları:

Herhangi bir elemanda meydana gelen σ gerilmesi, müsaade edilen σ_a emniyet gerilmesi değerini aşmamalı. Yani $|\sigma| \leq \sigma_a$, ($\sigma_a > 0$) olmalıdır. Mutlak değer, çekme veya basma durumu göz önünde bulundurularak kullanılmıştır. Denklem (2.6) 'dan F_1 daima negatiftir. Yani birinci elemanda basma gerilmesi oluşur. Eğer F_2 'de basma kuvveti olursa [yani denklem (2.7) 'de $(\sin \theta)/h \geq 2(\cos \theta)/s$] ve A_1 ile A_2 sırasıyla birinci ve ikinci elemanların kesit alanları ise;

$$A_1 = \frac{\pi}{4}(x_3^2 - x_4^2) \quad , \quad A_2 = \frac{\pi}{4}(x_5^2 - x_6^2)$$

$$\sigma_1 = \frac{F_1}{A_1} \leq \sigma_a \quad , \quad \sigma_2 = \frac{F_2}{A_2} \leq \sigma_a$$

olur ve bu ifadelerden hareketle dizayn sınırlamaları;

$$\bullet \quad g_1(x) \equiv \frac{2Wl}{\pi(x_3^2 - x_4^2)} \left[\frac{\sin \theta}{x_1} + \frac{2 \cos \theta}{x_2} \right] - \sigma_a \leq 0 \quad (2.9)$$

$$\bullet \quad g_2(x) \equiv \frac{2Wl}{\pi(x_5^2 - x_6^2)} \left[\frac{\sin \theta}{x_1} - \frac{2 \cos \theta}{x_2} \right] - \sigma_a \leq 0 \quad (2.10)$$

olacaktır.

Bunlara ek olarak dizayn değişkenleri için belirtilecek maksimum (x_{iu}) ve minimum (x_{il}) değerlerden oluşacak sınırlamalar;

$$\bullet \quad g_i(x) \equiv x_{il} \leq x_i \leq x_{iu} \quad i=1 \rightarrow 6 \quad (2.11)$$

olur. Dizayn değişkenleri için üst sınırlama olmayabilir, fakat negatif bir boyut mantıksız olacağından alt sınırlama olarak en azından sıfır veya sıfırdan büyük olduğunu belirtecek bir sınırlama bulunmalıdır. [1]

Denklem (2.10) 'da $(\sin \theta)/x_1 < 2(\cos \theta)/x_2$ olursa, F_2 çekme gerilmesidir. Bu yüzden ikinci eleman için yazılan (2.10) no'lu dizayn sınırlaması şu şekilde yazılır;

$$\bullet \quad g_2(x) \equiv \frac{-2Wl}{\pi(x_5^2 - x_6^2)} \left[\frac{\sin \theta}{x_1} - \frac{2 \cos \theta}{x_2} \right] - \sigma_a \leq 0 \quad (2.12)$$

2.2 Minimum

Optimizasyon problemi, hedef fonksiyonuna uygun bölge içinde minimum değerini veren dizayn değişkeni vektörünün (x^*) bulunmasıdır. Genel (global) ve yerel (local) olmak üzere iki tür minimum tanımı vardır. [1]

2.2.1 Genel (global) minimum

$f(x)$, n değişkenli bir fonksiyon olmak üzere, uygun bölge içerisinde bütün x değerleri için aşağıdaki şart sağlanıyorsa, x^* değişken vektörü için, bu fonksiyon genel minimum noktasındadır. [1]

$$f(x^*) \leq f(x) \quad (2.13)$$

2.2.2 Yerel (local) minimum

$f(x)$, n değişkenli bir fonksiyon olmak üzere, x^* noktasının çok küçük farklarla sağ ve solunda yer alan ve aşağıda N olarak tanımlanan x noktaları için (2.13) denklemi sağlanıyorsa, x^* değişken vektörü için, bu fonksiyon genel minimum noktasındadır.

$$N = \{ x | x \in S \quad \text{ve} \quad \|x - x^*\| < \delta \} \quad (2.14)$$

S : Uygun bölge

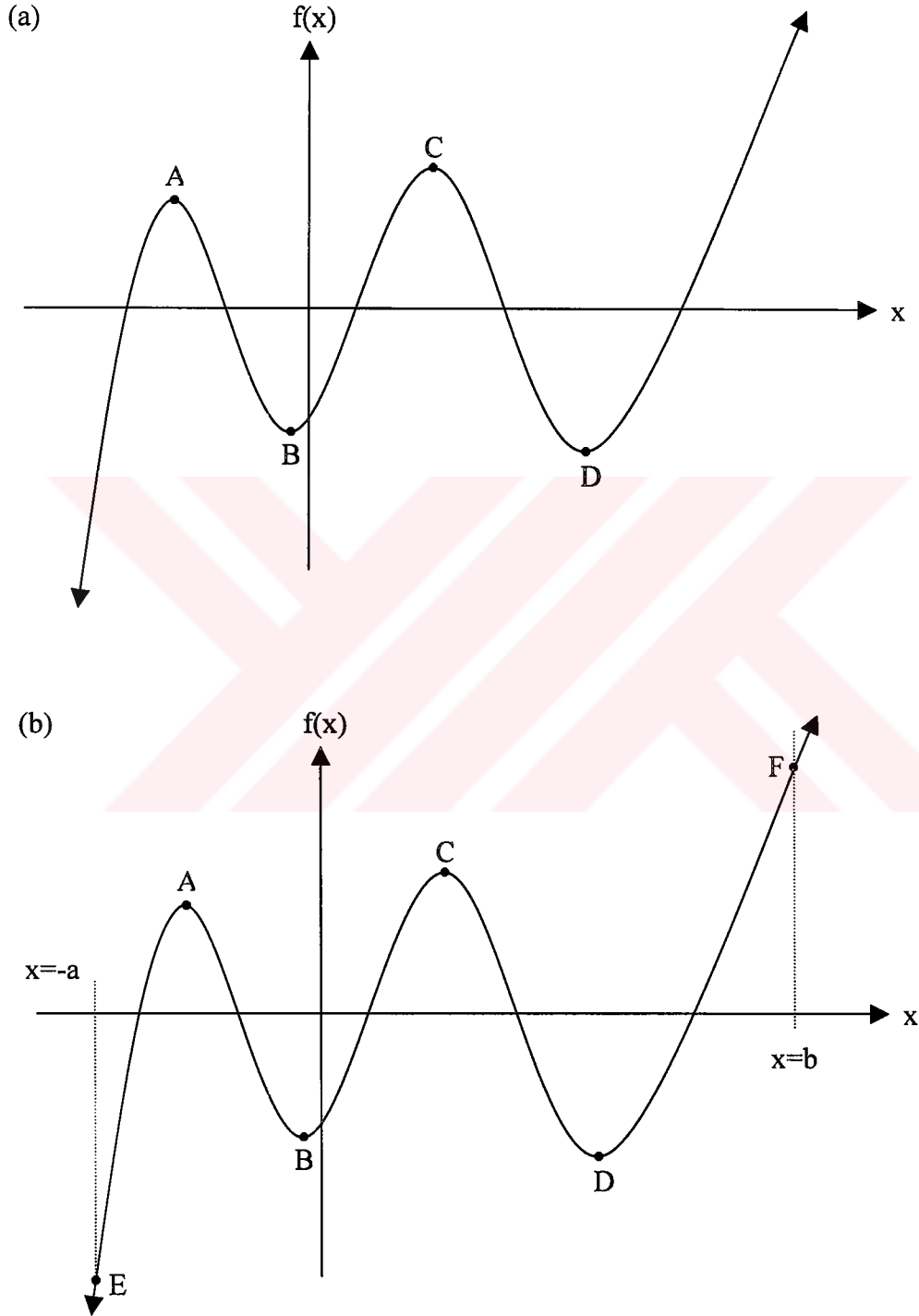
$$S = \{ x | h_j(x) = 0; j = 1 \rightarrow p; g_i(x) \leq 0; i = 1 \rightarrow m \}$$

δ : Çok küçük bir pozitif sayı.

Genel ve yerel minimumlar Şekil 2.5 'da ki grafiklerle daha iyi anlaşılabilir.

x değerinin $-\infty$ ile $+\infty$ arasında ($-\infty \leq x \leq \infty$) değerler alabildiği Şekil 2.5.(a) 'da, B ve D yerel minimum noktalarıdır. Çünkü bu noktaların çok az farklarla sağ ve solunda yer alan noktalar için fonksiyon daha büyük değerler almaktadır. Aynı şekilde A ve C noktaları da yerel maksimum noktalarıdır. Fakat fonksiyon sınırlanmadığı için fonksiyon için genel minimum ve maksimum yoktur. [1]

Eğer, Şekil 2.5.(b) 'de ki gibi x değerleri $-a$ ve b arasında sınırlanmışsa, E noktası genel minimum, F noktası ise genel maksimumdur.



Şekil 2.5- Optimum noktaların grafiksel gösterimi.(a) Sınırsız fonksiyon (genel minimum yok),(b) Sınırlı fonksiyon (genel ve yerel minimum ve maksimumlar var)

[1]

Genellikle çözümünü yapmadan önce bir problemin genel minimum noktasının olup olmadığını anlayamayız. Fakat bazı durumlar vardır ki çözümü yapılmassa da bir genel minimum kesinlikle vardır diyebiliriz. Weierstrass Teoremi işte bu tip durumları yorumlamaktadır.

2.2.3 Weierstrass teoremi

- ❖ Eğer $f(x)$ fonksiyonu uygun bölge (S) içerisinde sürekli ise ve S uygun seti kapalı ve sınırlı ise, $f(x)$ fonksiyonunun S içerisinde bir global minimumu vardır. Burada kapalı olmaktan kasıt, S 'nin kendisini sınırlayan bütün noktaları içermesidir.

Ele alacağımız bir örnekle bu teoremi daha iyi anlayabiliriz.

$$f(x) = -1/x \quad \text{ve} \quad S = \{x | 0 < x \leq 1\} \quad \text{olsun}$$

S uygun seti $x=0$ sınırını içermediği için kapalı değildir. Bu nedenle, her ne kadar $f(x)$, S 'de sürekli ise de, teoremin şartları sağlanmamış olur.

Eğer, $S = \{x | 0 \leq x \leq 1\}$ olarak tanımlanırsa, uygun bölge kapalı ve sınırlıdır. Ancak bu kez de $x=0$ için $f(x)$ tanımsız (sürekli) 'dir, yani teoremin şartları sağlanmamış olur. Yani her iki durum içinde kesinlikle bir genel minimum vardır diyemeyiz.

Bu noktada bir hususa dikkat etmek gerekir. Weierstrass teoreminin sağlanması genel minimum olduğunu garanti eder. Fakat sağlanmamış olması durumunda da genel minimum bulunması ihtimali vardır. [1]

2.2.4 Eğim (gradient) vektörü

Optimum dizayn bulma metotlarında kullanılan bu vektör, bir fonksiyonun birinci mertebeden kısmi türevlerini içermektedir. c veya ∇f sembolleri ile gösterilebilir. Örneğin n adet değişken içeren bir $f(x)$ fonksiyonunun x^* noktasındaki eğim vektörü aşağıdaki gibi gösterilebilir. [1]

$$\nabla f(x^*) = \begin{bmatrix} \frac{\partial f(x^*)}{\partial x_1} \\ \frac{\partial f(x^*)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x^*)}{\partial x_n} \end{bmatrix} = \left[\frac{\partial f(x^*)}{\partial x_1} \quad \frac{\partial f(x^*)}{\partial x_2} \quad \dots \quad \frac{\partial f(x^*)}{\partial x_n} \right]^T \quad (2.15)$$

2.2.5 Hessian matrisi

Eğim vektörünün her bir elemanının, bütün değişkenlere göre birer kez daha kısmi türevlerinin alınması ile elde edilen matristir. Bu matris ikinci mertebeden kısmi türevleri içerir. H veya $\nabla^2 f$ sembolleri ile gösterilebilir. Aşağıdaki gibi bir formu vardır.

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \quad (2.16)$$

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i} \quad i=1 \rightarrow n, j=1 \rightarrow n \text{ olduğundan Hessian daima simetrik bir}$$

matristir. [1]

2.3 Sınırlanmamış Fonksiyonların Yerel Minimumlarının Analitik Yolla Bulunması

Bir fonksiyonun yerel minimum noktası için *gerekli (necessary)* ve *yeter (sufficient)* olarak belirtilen durumlar araştırılır. Gerekli durumlar, minimum noktada mutlaka

sağlanmalıdır. Eğer gerekli durumlar herhangi bir noktada sağlanıyor ise, o nokta bir yerel maksimum veya dönüm noktası da olabileceğinden, sadece yerel minimum olmak için aday (candidate) hale gelir. Şayet yeter koşullar da sağlanıyorsa o nokta bir yerel minimumdur denilebilir. [1]

2.3.1 Tek değişkenli sınırlanmamış bir fonksiyonun yerel minimumu için gerekli ve yeter durumlar

2.3.1.1 Birinci mertebeden gerekli durum

Tek değişkenli sınırlanmamış bir fonksiyonun bir x^* noktasında aday yerel minimum olması için aşağıdaki durum gereklidir.

$$f'(x^*) = 0 \quad (2.17)$$

Birinci mertebeden diye isimlendirilmesinin nedeni fonksiyonun birinci türevini içermesidir. [1]

2.3.1.2 Yeter durumlar

Denklem (2.17) ile belirtilen gerekli durum x^* noktasında sağlanmışsa, yeterlilik için aşağıdaki durum incelenir.

$$f''(x^*) > 0 \quad (2.18)$$

Eğer bu durum da sağlanıyorsa x^* bir yerel minimum noktasıdır. [1]

2.3.1.3 İkinci Mertebeden gerekli durum

Denklem (2.18) 'in sağlanmaması x^* noktasının yerel minimum olmadığı anlamına gelmez. Aşağıda verilen ikinci mertebeden gerekli durum incelenir.

$$f''(x^*) \geq 0 \quad (2.19)$$

Eğer $f''(x^*) = 0$ ise gerekli durum sağlanmıştır ve yerel minimumu anlamak için daha yüksek mertebeden türevler alınır.

Gerekli durum olarak denklem (2.17) 'ye benzer şekilde $f'''(x^*) = 0$ olmalıdır. Yeter durum olarak ise denklem (2.18) 'e benzer şekilde $f^{IV}(x^*) > 0$ olmalıdır. Eğer gerekiyorsa işlem bu şekilde devam ettirilerek x^* noktasının bir yerel minimum olup olmadığı anlaşılabilir. [1]

2.3.2 Çok değişkenli sınırlanmamış bir fonksiyonun yerel minimumu için gerekli ve yeter durumlar

2.3.2.1 Gerekli durum

$f(x)$ fonksiyonunun x^* noktasında minimum olması için aşağıdaki durum gereklidir.

$$\nabla f(x^*) = \frac{\partial f(x^*)}{\partial x_i} = 0 \quad i=1 \rightarrow n$$

2.3.2.2 İkinci mertebeden gerekli durum

x^* noktasındaki Hessian matrisi pozitif yarı-belirli (positive semidefinite) veya pozitif belirli (positive definite) olmalıdır.

$$H(x^*) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right]_{(n \times n)}$$

Bir matrisin pozitif yarı-belirli veya pozitif belirli olup olmadığı özdeğerlerinin

(eigenvalues) bulunması ile anlaşılabilir. Örneğin $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ matrisinin

özdeğerleri (λ) şöyle bulunur.

$$\begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = 0 \quad (\text{Köşegendeki her terimden } \lambda \text{ çıkarılır.})$$

$$\lambda^2 - 4\lambda + 3 = 0 \quad \square \quad \lambda_1 = 3, \lambda_2 = 1$$

- Eğer bütün özdeğerler pozitif ise matris pozitif belirli ($\lambda_i > 0$, $i=1 \rightarrow n$)
- Eğer bütün özdeğerler negatif değilse ($\lambda_i \geq 0$, $i=1 \rightarrow n$) matris pozitif yarı-belirlidir. Pozitif yarı-belirli olması için özdeğerlerden en az birinin sıfır olması gerekir.

2.3.2.3 İkinci mertebe yeter durum

Eğer $H(x^*)$ pozitif belirli ise, $f(x)$ fonksiyonu x^* noktasında yerel minimumdur.

2.4 Sınırlanmış Fonksiyonların Yerel Minimumlarının Analitik Yolla Bulunması

2.4.1 Lagrange çarpanları (Lagrange multipliers)

Gerek eşitlik gerekse eşitsizlik halinde olsun, her türlü sınırlamanın kendisi ile ilgili olan çarpanları vardır. Bunlara Lagrange çarpanları denir. Lagrange çarpanları optimizasyon teorisinde önemli bir rol oynarlar. Bunların değerleri hedef ve sınırlama fonksiyonlarının formuna bağlıdır. Bu fonksiyonların değişmesiyle Lagrange çarpanları da değişir. Bu çarpanlar için eşitlik sınırlamalarında v , eşitsizlik sınırlamalarında ise u sembolleri kullanılacaktır. v herhangi bir reel sayı olabilir. u negatif olmayan ($u \geq 0$) bir sayı olmalıdır. Bunlara bağlı olarak Lagrange denklemi aşağıdaki gibidir. [1]

$$L(x, u, v) = f(x) + \sum_{i=1}^p v_i h_i(x) + \sum_{i=1}^m u_i g_i(x) \quad (2.20)$$

2.4.2 Kuhn-Tucker gerekli durumları

Denklem (2.20) 'den hareketle bir x^* noktası için aşağıdaki durumlar sağlanıyor ise, x^* noktası bir aday minimum noktasıdır.

$$\frac{\partial L}{\partial x_j} \equiv \frac{\partial f}{\partial x_j} + \sum_{i=1}^p v_i \frac{\partial h_i}{\partial x_j} + \sum_{i=1}^m u_i \frac{\partial g_i}{\partial x_j} = 0 \quad j = 1 \rightarrow n \quad (2.21)$$

$$g_i(x^*) \leq 0 \quad i = 1 \rightarrow m \quad (2.22)$$

$$u_i^* g_i(x^*) = 0 \quad i = 1 \rightarrow m \quad (2.23)$$

$$u_i \geq 0 \quad i = 1 \rightarrow m \quad (2.24)$$

Denklem (2.23) 'den hareketle u_i veya g_i 'den bir tanesinin sıfır olması gerektiği için bununla ilgili bütün durumlar incelenir [1]. Bir örnekle bu ifadeler daha iyi anlaşılabilir.

➤ $f(x) = x_1^2 + x_2^2 - 2x_1 - 2x_2$ fonksiyonunu

- $g_1 \equiv -2x_1 - x_2 + 4 \leq 0$
- $g_2 \equiv -x_1 - 2x_2 + 4 \leq 0$ sınırlamalarına göre minimum yap.

Denklem (2.20)'den Lagrange denklemi oluşturularak gerekli durumlar incelenir.

$$L = x_1^2 + x_2^2 - 2x_1 - 2x_2 + u_1(-2x_1 - x_2 + 4) + u_2(-x_1 - 2x_2 + 4)$$

$$\frac{\partial L}{\partial x_1} \equiv 2x_1 - 2 - 2u_1 - u_2 = 0 \quad (a)$$

$$\frac{\partial L}{\partial x_2} = 2x_2 - 2 - u_1 - 2u_2 = 0 \quad (b)$$

$$g_1 \equiv -2x_1 - x_2 + 4 \leq 0 \quad (c)$$

$$g_2 \equiv -x_1 - 2x_2 + 4 \leq 0 \quad (d)$$

$$u_1 g_1 = 0 \quad , \quad u_2 g_2 = 0 \quad (e)$$

$$u_1 \geq 0 \quad , \quad u_2 \geq 0 \quad (f)$$

(e) ile belirtilen durumlar için aşağıdaki ihtimaller vardır.

1. $u_1 = 0, \quad u_2 = 0$
2. $u_1 = 0, \quad g_2 = 0$
3. $g_1 = 0, \quad u_2 = 0$
4. $g_1 = 0, \quad g_2 = 0$

Bu ihtimaller için (a), (b), (c), (d), (e) ve (f) durumları göz önüne alınıp oluşan denklemler çözülürse şu sonuçlar elde edilir.

□ Durum 1. $u_1 = 0, \quad u_2 = 0$ için;

$x_1 = x_2 = 1$ olur fakat

$g_1 = g_2 = 1$ olup (c) ve (d) ihlal edildiğinden bu uygun bir dizayn olmayacaktır.

□ Durum 2. $u_1 = 0, \quad g_2 = 0$ için;

$x_1 = 1.2, \quad x_2 = 1.4, \quad u_2 = 0.4, \quad f = 0.2$ olur fakat

$g_1 = 0.2$ olup (c) ihlal edildiğinden bu da uygun bir dizayn olmayacaktır.

□ Durum 3. $g_1 = 0, \quad u_2 = 0$ için;

$x_1 = 1.4, \quad x_2 = 1.2, \quad u_1 = 0.4, \quad u_2 = 0, \quad f = 0.2$ olur fakat

$g_2 = 0.2$ olup (d) ihlal edildiğinden bu da uygun bir dizayn olmayacaktır.

□ Durum 4. $g_1 = 0, \quad g_2 = 0$ için;

$x_1 = 4/3, \quad x_2 = 4/3, \quad u_1 = 2/9, \quad u_2 = 2/9, \quad f = 0.222$ olur ve bu durum için bütün şartlar sağlandığından $x^* = (4/3, 4/3)$ bir aday yerel minimum noktasıdır. [1]

Sınırlanmış problemler için yeter durum bölüm 2.4.5. de anlatılacaktır.

2.4.3 Genel optimum

Optimum dizayn problemlerinde genel optimumun nasıl bulunacağı sorusuna iki şekilde cevap verilebilir.

1. Eğer Weierstrass teoremi sağlanıyorsa, bir genel minimumun olduğu kesinlik kazanır. Bu durumda bütün aday optimum noktalar hesaplanır ve bunlardan hedef fonksiyonuna en küçük değerini veren nokta seçilir.
2. Eğer problemin konveks olduğu gösterilirse, bulunan herhangi bir yerel minimum aynı zamanda genel minimumdur. [1]

2.4.4 Fonksiyon ve setlerin konveksliği

S seti, genel optimizasyon sınırlamaları ile tanımlanan bir set olsun.

$$S = \{x \mid g_i(x) \leq 0, i = 1 \rightarrow m; h_j(x) = 0, j = 1 \rightarrow p\} \quad (2.25)$$

Eğer bütün g_i eşitsizlik sınırlaması fonksiyonları konveks ve h_i eşitlik sınırlaması fonksiyonları doğrusal (linear) ise S seti konvektir.

S konveks seti içerisinde tanımlanan n değişkenli bir $f = (x_1, x_2, \dots, x_n)$ fonksiyonunun Hessian matrisi, S içindeki bütün noktalarda pozitif belirli veya pozitif yarı belirli ise, bu f fonksiyonu konvektir. Eşitsizlik sınırlaması fonksiyonlarının konveksliği de aynı kriterle göre anlaşılabilir.

Doğrusal olmayan (nonlinear) eşitlik sınırlaması fonksiyonları daima konveks değildir (nonconvex).

Doğrusal olan eşitsizlik sınırlamaları konvektir.

- ❖ Eğer yukarıda belirtilen kriterlere göre konveks olan bir S seti içerisinde, hedef fonksiyonu da konveks ise problem konvektir. Böyle bir problem için, daha önce açıklanan Kuhn-Tucker durumlarını sağlayan herhangi bir yerel minimum noktası aynı zamanda bir genel minimumdur.

Fakat burada bir hususu da belirtmek gerekir ki, konveks olmayan problemler de yine genel minimuma sahip olabilir. [1]

2.4.5 Sınırlanmış fonksiyonlar için ikinci mertebe durumlar

Bu bölümde gerekli koşulları sağlayarak aday yerel minimum noktası olan çözümlerin, yeter durumları ve ikinci mertebeden gerekli durumları incelenerek bir aday dizaynın gerçekten bir yerel minimum olup olmadığı belirlenecektir. İlk olarak bir önceki bölümde açıklanan konveks problemler için daha sonra genel optimizasyon problemleri için yeter durumlar incelenecektir.

2.4.5.1 Konveks problemler için yeter durumlar

Eğer $f(x)$, konveks bir set içerisinde tanımlanan konveks bir fonksiyon ise Kuhn-Tucker durumlarına bakılır. Bulunan herhangi bir nokta genel minimumdur. Bu durumu eşitlik sınırlaması olmayan küçük bir örnekle açıklayalım:

$$\begin{aligned} &\triangleright f(x) = (x_1 - 1.5)^2 + (x_2 - 1.5)^2 \\ &\bullet g(x) = x_1 + x_2 - 2 \leq 0 \end{aligned}$$

Yukarıdaki problemde $g(x)$ doğrusal olduğu için set konvektir. Hedef fonksiyonunun Hessian matrisi ve bu matrise ait özdeğerler aşağıdaki gibidir.

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad \lambda_1 = 2, \quad \lambda_2 = 2$$

H matrisi, özdeğerlerinin her ikisi de pozitif olduğundan pozitif belirlidir. Dolayısıyla $f(x)$ de konvektir. Hem sınırlama seti hem de $f(x)$ konveks olduğu için problem konvektir. O halde, Kuhn-Tucker durumları incelenerek elde edilecek olan $x_1 = 1$, $x_2 = 1$ ve $u=1$ çözümünden;

$x_1 = 1$, $x_2 = 1$ problem için genel minimumdur denilebilir. [1]

2.4.5.2 Genel problemler için ikinci merteye durumlar

2.4.5.2.1 Kuvvetli yeter durum

x^* Kuhn-Tucker gerekli durumlarını sağlamış olsun. Lagrange denkleminin aşağıdaki gibi ikinci merteden kısmi türevleri alınarak oluşturulan Hessian matrisi, x^* noktasında pozitif belirli ise bu nokta bir yerel minimum noktasıdır.

$$\nabla^2 L = \nabla^2 f + \sum_{i=1}^p v_i^* \nabla^2 h_i + \sum_{i=1}^m u_i^* \nabla^2 g_i \quad (2.26)$$

Eğer Hessian matrisi pozitif belirli değilse bir sonraki bölümde belirtilen bir d vektörü hesaplanarak istenen yeter durum incelenir. [1]

2.4.5.2.2 Yeter durumlar

x^* , Kuhn-Tucker gerekli durumlarını sağlayan bir nokta olsun. d , bu noktadan itibaren sınırlama gradyenlerine (∇h_i ve ∇g_i) dik olarak uygun yönlerde hareketi ifade eden sıfırdan farklı ($d \neq 0$) bir vektör olsun. (2.27) ve (2.28) denklemleri kullanılarak d vektörü hesaplanır. Daha sonra (2.29) denklemi ile belirtilen durumun sağlanıp sağlanmadığına bakılır. Eğer sağlanıyorsa x^* , bir yerel minimum noktasıdır.

$$\nabla h_i^T d = 0 \quad i=1 \rightarrow p \quad (2.27)$$

$$\nabla g_i^T d = 0 \quad i=1 \rightarrow m \text{ (} u_i > 0 \text{ olan bütün aktif eşitsizlikler (} g_i(x)=0 \text{) için)} \quad (2.28)$$

$$Q = d^T \nabla^2 L(x^*) d \quad \text{olmak üzere} \quad Q > 0 \quad (2.29)$$

2.5 Sınırlanmamış Problemlerin Optimum Dizaynı için Sayısal Yöntemler

Bazı optimizasyon problemlerinde hedef fonksiyonu veya sınırlama fonksiyonlarının bir veya birden fazlasının doğrusal olmama durumu olabilir. Bu tip problemlerin çözümü için sayısal yöntemler gerekir. Çünkü daha önceki bölümlerde belirtilen

analitik metotlarla, bu tip problemleri çözmenin bazı zorlukları vardır. Bu zorluklardan önemli olan iki tanesi şunlardır:

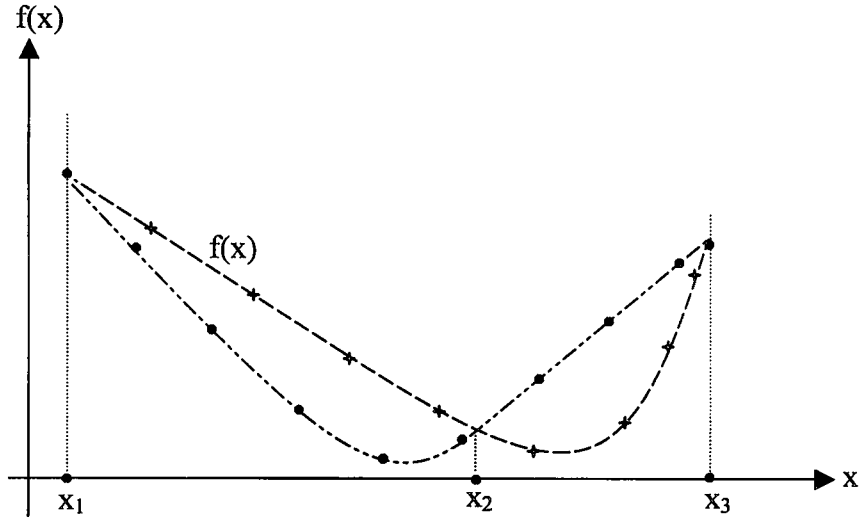
1. Çok sayıda dizayn değişkeni ve sınırlama olabilir. Gerekli koşullar arandığında çok sayıda denklem oluşur ki çözümü zordur. Bu sorun özellikle sınırlanmış problemlerde kendini belli eder.
2. Hedef ve sınırlama fonksiyonlarının dereceleri çok yüksek olabilir. Bu tip durumlarda değişken sayısı az olsa dahi, gerekli durumlar yüksek dereceli olacağından yine çözüm güçleşir. [1]

Bu nedenlerden dolayı oluşturulan problemin dizayn değişkenlerine mantıklı olabilecek tahmini başlangıç değerleri verilir. Daha sonra belirli bir sistematığı olan sayısal yöntemler kullanılarak başlangıç değerleri değiştirilmek suretiyle optimum dizayna ulaşılmaya çalışılır. Değişik zamanlarda geliştirilmiş ve bazıları diğerlerinden daha iyi olan birçok sayısal yöntem vardır.

Sayısal yöntemlerin, analitik yöntemlerden farklı kavramları vardır. Analitik yaklaşımda, gerekli ve yeter durumlar ışığında çözüm yapılarak yerel minimuma ulaşılır. Fakat sayısal yöntemlerde, bir başlangıç dizaynı oluşturulup, iteratif değişimlerle optimum olma durumları sağlanana kadar bu dizayn değiştirilerek çözüm yapılır. Birden fazla iterasyon gerekebilir. Bu bölümde sınırlanmamış problemler için kullanılan sayısal yöntemlerden popüler olan birkaç tanesinden bahsedilecektir. [1]

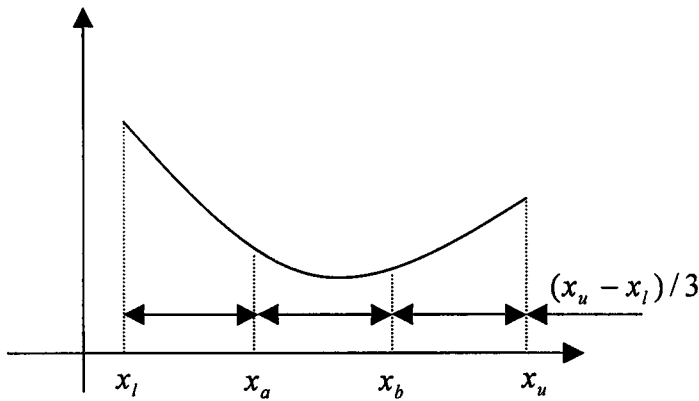
2.5.1 Altın bölge metodu (Golden section method)

Bu yöntemde tek değişkenli bir fonksiyon için, bir minimum bulunduğu kesin olarak belirlenen aralık (interval) seçilir. Bu aralık her seferinde 1/3 oranında olmak üzere istenen kesinlik (accuracy) sağlanana kadar daraltılarak sonuca ulaşılır.



Şekil 2.6- Bir minimumun kesin olarak bulunduğu aralığın tespiti

Yukarıdaki şekilde görüldüğü gibi bir x_1 noktası rasgele seçilerek buna karşılık gelen fonksiyon değeri ($f(x_1)$) hesaplanır. Daha sonra aynı şekilde bir x_2 noktası seçilir ve buna ait fonksiyon değeri olan $f(x_2)$ hesaplanır. Elde edilen iki fonksiyon değeri kıyaslanır. Eğer, $f(x_2) < f(x_1)$ ise bu yönde gidildiğinde fonksiyonda bir azalma olduğu dolayısıyla bir minimum noktaya doğru gidildiği anlaşılır. Aynı yönde ilerleyip bir x_3 noktası seçilerek yine buna ait $f(x_3)$ hesaplanır. Eğer, $f(x_3) < f(x_2)$ ise ilerlemeye devam edilir ve $f(x_n) > f(x_{n-1})$ durumu meydana geldiği anda durulur. Şekilde $f(x_3) > f(x_2)$ olduğundan bu noktada durulur. x_2 noktasından geçen fonksiyon farklı çizgilerle belirtildiği gibi iki farklı yol izleyebileceğinden, minimumun x_1 ile x_2 arasında mı, yoksa x_2 ile x_3 arasında mı olduğu anlaşılamaz, fakat x_1 ile x_3 arasında olduğu kesinlik kazanır.



Şekil 2.7- Belirlenen aralığın üç eşit parçaya ayrılması [1]

Bu iki nokta alt ve üst sınırlar olduklarını belirtmek üzere, $x_1 = x_l$ ve $x_3 = x_u$ olarak yeniden isimlendirilirler. Buna göre aralığın büyüklüğü, $I = x_u - x_l = x_3 - x_1$ olur. Daha sonra bu aralık Şekil 2.7 de ki gibi üç eşit parçaya bölünür. x_a ve x_b noktaları aşağıdaki formüllerle hesaplanır.

$$x_a = x_l + \frac{1}{3}(x_u - x_l) = \frac{1}{3}(x_u + 2x_l) \quad (2.30)$$

$$x_b = x_l + \frac{2}{3}(x_u - x_l) = \frac{1}{3}(2x_u + x_l) \quad (2.31)$$

Bu noktada iki farklı durum oluşabilir:

1. Eğer $f(x_a) < f(x_b)$ ise, minimum x_l ve x_b arasındadır. x_b ve x_u arasındaki üçüncü bölüm iptal edilir. Aralık için yeni sınırlar $x_l = x_l$ ve $x_u = x_b$ olur. Yeni belirlenen bölge, aynı işlemler tekrar edilerek istenen kesinlik sağlanmaya çalışılır. Kesinlik, genelde 0.001 olarak alınabilir. Aralığın büyüklüğü (I) bu değere veya daha altına düştüğünde istenen kesinlik sağlanmış olur.
2. Eğer $f(x_a) > f(x_b)$ ise, minimum x_a ve x_u arasındadır. x_l ile x_a arasındaki bölme iptal edilir. Aralık için yeni sınırlar $x_l = x_a$ ve $x_u = x_u$ olur. Yeni aralık bu şekilde daraltılarak istenen kesinlik sağlanır. [1]

2.5.2 Dikey azalma metodu (Steepest descent method)

Bu metotta ilk olarak keyfi bir nokta (X_q) seçilir. Bu noktadan başlanıp bir yön vektörü yardımı ile istenen yönlerde ilerlenerek, bir yerel minimum noktaya ulaşılmaya çalışılır. Daha sonra geline noktasından itibaren yeni bir yönde ilerlenerek bir başka yerel minimuma ulaşılır. Bu şekilde yapılan birkaç denemeden sonra genel minimuma ulaşılır. Matematik olarak bir X_q noktasından X_{q+1} noktasına yapılan hareket n boyutlu bir uzayda şöyle ifade edilir. [4]

$$X_{q+1} = X_q + \alpha.d_q \quad (2.32)$$

$f(X_{q+1})$ bir yerel minimumdur. Formülde geçen d_q ifadesi, arama yönünü belirleyen vektördür. Bu vektörün değişken sayısına (n) bağlı olarak n elemanı vardır. Örneğin; üç boyutlu uzayda, $d_q = \{a \ b \ c\}^T$, x ekseninde a birim, y ekseninde b birim, z ekseninde c birim ilerlemeyi ifade etmektedir.

Bu metotta arama işlemi değişmeli olarak yapılır. d vektörü $d_0 = \{1 \ 0 \ 0 \ \dots\}^T$, $d_1 = \{0 \ 1 \ 0 \ \dots\}^T$, $d_2 = \{0 \ 0 \ 1 \ \dots\}^T$ gibi değerler alır. Arama yönü bir kere seçilip bu yöndeki hedef fonksiyonun yerel minimumu bulunur ki bu nokta bir α^* değerine doğru yığılır. Daha sonra (2.32) denklemi ile başlangıç noktası taşınır. Aynı işlemler yeni bir arama yönü vektörü seçilip tekrar edilerek genel minimuma ulaşılır. Bir örnekle bu metodu daha iyi anlayabiliriz. [4]

$$\triangleright f(x) = x_1^2 - 2x_1x_2 + 2x_2^2 - 3x_1 + 6$$

Bu fonksiyonun minimumunu bulmak için $X_0 = \{0 \ 0\}^T$ noktasından başlayıp arama yaparak birkaç adımda sonuca ulaşılabilir.

Adım 1. $d = \{1 \ 0\}^T$ yönünde arama için x_2 sabit kalacaktır. Bu yüzden hedef fonksiyonu $f(x) = x_1^2 - 3x_1 + 6$ şeklinde yazılabilir. Bu fonksiyonun yerel minimum noktasını daha önceki bölümlerde bahsi geçtiği şekli ile gerekli ve yeter durumlar incelenerek $\alpha^* = 1.5$ olarak bulunur ve bundan hareketle yeni arama noktası belirlenir.

$$X_1 = \{0 \ 0\}^T + 1.5\{1 \ 0\}^T = \{1.5 \ 0\}^T$$

Adım 2. $d_1 = \{0 \ 1\}^T$ için

$$f(x) = 3.75 - 3x_2 + 2x_2^2 \quad \rightarrow \quad \alpha^* = 0.75$$

$$X_2 = \{1.5 \ 0\}^T + 0.75\{0 \ 1\}^T = \{1.5 \ 0.75\}^T$$

Adım 3. $d_2 = \{1 \ 0\}^T$ için

$$\alpha^* = 0.75$$

$$X_3 = \{1.5 \ 0.75\}^T + 0.75\{1 \ 0\}^T = \{2.25 \ 0.75\}^T$$

Adım 4. $d_3 = \{0 \ 1\}^T$ için

$$\alpha^* = 0.375$$

$$X_4 = \{2.25 \quad 1.125\}^T$$

Bu adımlar devam ettirilerek genel minimum noktaya ulaşılır. Bu nokta $\{3 \ 1.5\}$ tur. Kuhn-Tucker durumları ile analitik yolla da aynı sonuca ulaşılabilir. [4]

2.6 Sınırlanmış Problemlerin Optimum Dizaynı için Sayısal Yöntemler

Sınırlanmış problemlerin çözümü için direk ve dolaylı olmak üzere iki farklı yol vardır. Dolaylı çözüm yöntemlerinde, sınırlama fonksiyonları hedef fonksiyonuna bazı yöntemlerle dahil edilip, sınırlanmamış problemlere uygulanan tekniklerle çözülür. Direk yöntemlerde ise hedef ve sınırlama fonksiyonları ayrı tutularak, direk çözüm metotları ile sonuca ulaşılır.

2.6.1 Dolaylı çözüm metotları

Sınırlama fonksiyonlarını hedef fonksiyonuna eklemek için penaltı fonksiyonları veya Lagrange çarpanları kullanılır. Penaltı fonksiyonları dış (exterior) ve iç (interior) olmak üzere iki tiptedir. Dış penaltı fonksiyonunda minimuma uygun bölgenin dışından, iç penaltı fonksiyonunda ise uygun bölgenin içerisinde yaklaşılar.

2.6.1.1 Dış penaltı fonksiyonu metodu

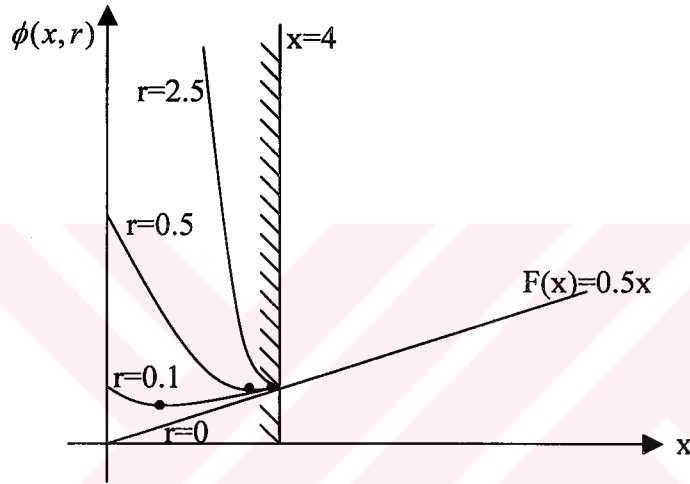
Hem iç hem de dış penaltı fonksiyonları $P(X)$, bir r_i sayısı ile çarpılıp, hedef fonksiyonuna eklenerek, hedef fonksiyonu aşağıdaki forma getirilir.

$$\phi(X, r_i) = F(X) + r_i P(X) \quad (2.33)$$

p tane eşitsizlik, m tane eşitlik sınırlaması olan bir dizayn probleminde dış penaltı fonksiyonu aşağıdaki gibidir.

$$P(X) = \sum_{j=1}^p [\max\{0, g_j(X)\}]^2 + \sum_{k=1}^m [h_k(X)]^2 \quad (2.34)$$

Bu fonksiyon sadece ihlal edilen sınırlama olduğunda değer almaktadır. Uygun bölge içerisinde $\phi(x) = f(x)$ 'tir. Pozitif r sayısı küçük bir değerden başlayarak artırılır. Bunun için $r_{i+1} / r_i = 5$ gibi bir artım kullanılabilir. Aşağıdaki şekilde dış penaltı fonksiyonu yaklaşımına basit bir örnek gösterilmiştir.



Şekil 2.8- $F(x)=0.5.x$ fonksiyonunun $4 - x \leq 0$ sınırlamasına göre dış penaltı fonksiyonu [3]

Dış penaltı fonksiyonlarının avantaj ve dezavantajları:

1. Genel sınırlanmış problemlere, yani hem eşitlik hem de eşitsizlik sınırlamaları içeren problemlere uygulanabilir.
2. Başlangıç noktası rasgele seçilebilir.
3. Metot uygun olmayan bölgede iterasyon yaptığından problem fonksiyonları tanımsız olabilir.
4. Eğer iterasyon işlemi erken kesilirse sonuç dizaynı uygun olmayabilir.

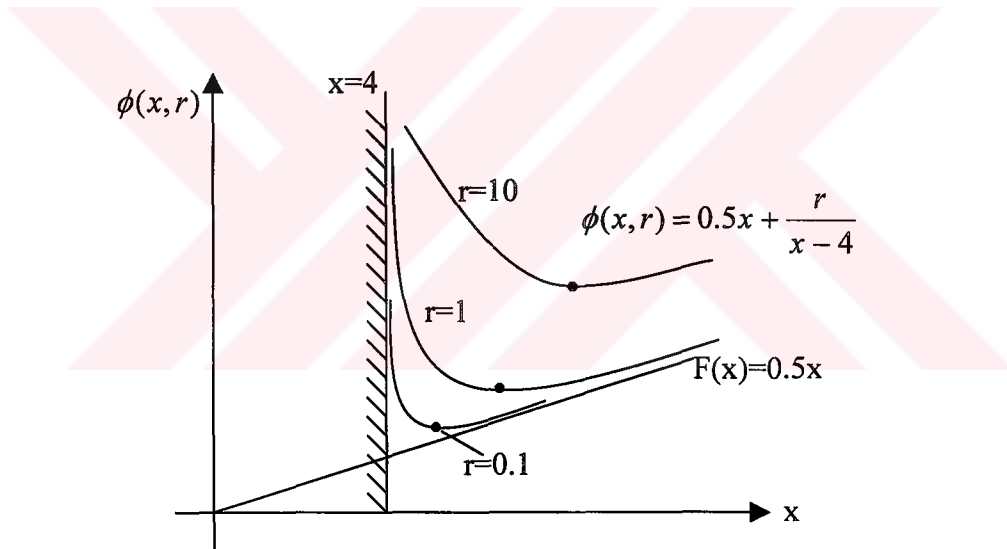
Şekil 2.8 'den görüldüğü gibi r büyüdükçe ϕ fonksiyonunun minimumu sınırlamaya doğru yaklaşmaktadır. Fakat r sayısının çok büyük değerleri için sayısal zorluklar ortaya çıkar ve tam olarak minimumu bulmak mümkün değildir. [1,3,4]

2.6.1.2 İç penaltı fonksiyonu metodu

İç penaltı fonksiyonu kullanıldığında minimum noktaya yaklaşım uygun bölgenin içinden sağlanır. Bu yüzden doğru minimum bulunmasa da dizayn en azından uygun ‘dur. Yaygın bir iç penaltı fonksiyonu tipi olan, ters iç penaltı fonksiyonu aşağıdaki gibidir.

$$P(x) = r \sum_{j=1}^p 1/g_j(x) \quad (2.35)$$

Bu tip bir yaklaşımda, dıştan yaklaşımdan farklı olarak r sayısı azaldıkça minimum noktaya yaklaşılır. Aşağıdaki şekilde bu tip bir yaklaşım gösterilmiştir.



Şekil 2.9- $F(x)=0.5.x$ fonksiyonunun $4 - x \leq 0$ sınırlamasına göre iç penaltı fonksiyonu [3]

Kullanılan bir başka iç penaltı fonksiyonu da logaritmik iç penaltı fonksiyonudur.

$$\phi(x, r) = f(x) - r \sum_{j=1}^p \log(g_j(x)) \quad (2.36)$$

İç penaltı fonksiyonu metotları aynı zamanda bariyer fonksiyonu metotları olarak da adlandırılır. Bunun nedeni uygun bölgeden iterasyonlara başlanıp, uygun olmayan bölgeye geçilememesidir.

İç penaltı fonksiyonlarının avantaj ve dezavantajları:

1. Sadece eşitsizlik sınırlamalı problemlere uygulanabilir.
2. Başlangıç dizaynı uygun olmalıdır.
3. Metot daima uygun bölgede ilerlemeyi sağladığından, işlem erken kesilse bile sonuç dizaynı uygun olur.

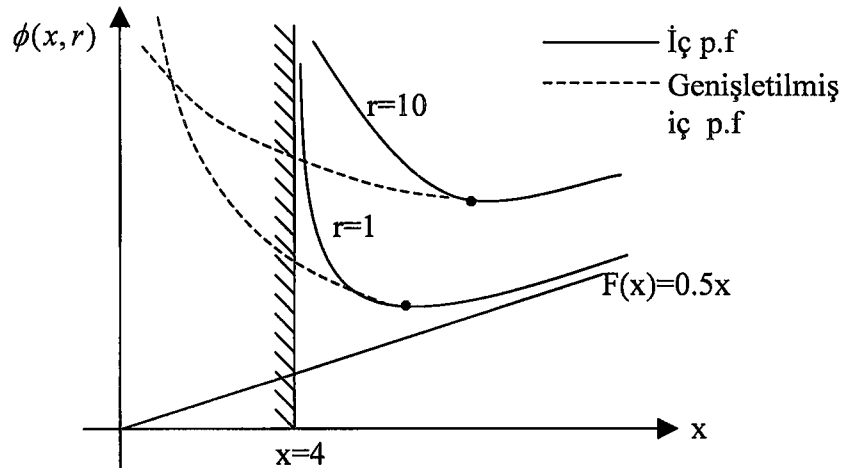
İç ve dış penaltı fonksiyonları arasında bir kıyaslama yaptığımızda, her ne kadar iç penaltı fonksiyonu metodunda uygun bölgede ilerlense de, başlangıç dizaynının uygun olması şartı vardır. Bu nedenle her iki metodun birleştirilmiş bir hali olan, genişletilmiş iç penaltı fonksiyonu kullanılabilir. İkinci dereceden bir iç penaltı fonksiyonu aşağıdaki gibidir. [1,3]

$$\phi(x, r) = f(x) + r \sum_{j=1}^p p(g_j) \quad (2.37)$$

burada,

$$p(g_j) = \begin{cases} 1/g_j & g_j \geq g_0 \text{ için} \\ 1/g_0 [3 - 3(g_j/g_0) + (g_j/g_0)^2] & g_j < g_0 \text{ için} \end{cases} \quad (2.38)$$

$$g_0 = cr^{1/2}$$



Şekil 2.10- Genişletilmiş iç penaltı fonksiyonu [3]

2.6.1.3 Artırmalı Lagrange çarpanları metodu

Penaltı fonksiyonu metodlarında çözüm r sayısının seçimine bağlıdır ve kesin bir optimum çözüm bulmak imkansızdır. Lagrange çarpanlarını kullanarak verilen sınırlamalara göre tam olarak minimumu bulmak mümkündür.

Eşitlik sınırlamaları için artırmalı Lagrange denklemi (A) aşağıdaki gibidir.

$$L(x, \lambda) = f(x) + \sum_{j=1}^m \lambda_j h_j(x) \quad (2.39)$$

$$A(x, \lambda, r_i) = L(x, \lambda) + \sum_{k=1}^m r_i [h_k(x)]^2 \quad (2.40)$$

r sayısının sonlu değerleri için, λ değerleri değiştirilerek minimum noktaya ulaşılabilir.

Eşitsizlik sınırlamaları için artırmalı Lagrange denklemi (A) aşağıdaki gibidir.

$$L(x, \lambda) = f(x) + \sum_{j=1}^m \lambda_j \psi_j \quad (2.41)$$

$$A(x, \lambda, r) = L(x, \lambda) + \sum_{j=1}^m \psi_j^2 \quad (2.42)$$

$$\psi_j = \max[g_j(x), -\lambda_j / 2r_i]$$

λ_j çarpanları her bir iterasyonda aşağıdaki formülle yeniden tanımlanır. [4]

$$\lambda_j^{q+1} = \lambda_j^q + 2r_q \psi_j \quad (2.43)$$

2.6.2 Direk çözüm metotları

Bu tip çözüm yöntemlerinde hedef fonksiyonu ve sınırlama fonksiyonları birbirinden ayrı tutularak çözüm yoluna gidilir. Bir optimizasyon problemini direk yöntemlerle

özmek daha karmaşık hesaplamalar içerir. Daha önce geliştirilmiş bir çok yöntem vardır. Bu yöntemlerin bazıları şunlardır.

- Sınırlanmış dikey azalma metodu (Constrained steepest descent)
- Sınırlanmış Quasi-Newton metodu
- Uygun yönler metodu (Feasible directions)
- Eğim izdüşümü metodu (Gradient projection)



BÖLÜM 3

ANSYS PROGRAMI İLE DİZAYN OPTİMİZASYONU

3.1 Giriş

Optimizasyon modülü (/OPT), ANSYS programının içerdği modüllerden birisidir. Bu program yardımı ile optimum dizayn problemleri çözülebilmektedir. ANSYS programı altproblem yaklaşımı metodu (subproblem approximation method) ve birinci mertebe metodu (first order method) olmak üzere, iki farklı metot kullanarak dizayn optimizasyonu yapmaktadır. Altproblem yaklaşımı metodu çokça kullanılmakla birlikte, birinci mertebe metodu yüksek kesinlik gerektiren problemler için daha uygundur. Her iki metotta da program, analiz-dönüştürme-tanımlama işlemlerini içeren döngüleri (loop) gerçekleştirir. Bu döngü istenen kriterler sağlanana kadar devam ettirilir. [2]

3.2 ANSYS Terminolojisinde Optimizasyon için Kullanılan Temel Kavramlar

Dizayn optimizasyonu prosedürüne geçmeden önce, bazıları daha önce Bölüm 1’de anlatılan “design variable”, “state variable”, “objective function”, “feasible and infeasible designs”, “the analysis file”, “iterations”, “loops”, “optimization database” gibi ANSYS terminolojisinde geçen bazı ifadelerin tanımlarını yapacağız.

3.2.1 Dizayn değişkenleri (DVs : Design variables)

Bölüm 2 'de "dizayn değişkenleri" olarak tanımladığımız bu ifadelerden ANSYS’ de 60 taneye kadar tanımlanabilir. Kısaca DVs olarak isimlendirilebilir. [2]

3.2.2 Durum deęiřkenleri (SVs :State variables)

Bölüm 2' de "dizayn sınırlamaları" dedięimiz bu ifadelerden ise 100 taneye kadar tanımlanabilir. Her ne kadar Türkçe'ye durum deęiřkenleri olarak çevirmek uygun olsa da önceki bölümle uygunluk saęlaması bakımından bundan sonra da dizayn sınırlamaları olarak tanımlayacaęız. Kısaca SVs olarak isimlendirilebilir. [2]

3.2.3 Hedef fonksiyonu (objective function)

Bölüm 2 'de de yine aynı isimle tanımlanan bu ifadeden bir adet tanımlanır. Daha önce de belirtildięi gibi, uygun dizaynlar arasında hangisinin daha iyi olduęunun anlařılmasına yardım eden kriterdir.

DVs, SVs ve hedef fonksiyonu hep birlikte “optimizasyon deęiřkenleri” (optimization variables) olarak adlandırılırlar. Bir ANSYS optimizasyonunda bu deęiřkenler kullanıcı tarafından isimlendirilen ve “parametre” denilen deęiřkenlerle temsil edilir. Kullanıcı hangi parametreleri DVs, hangilerinin SVs ve hangisinin hedef fonksiyonu olacaęını belirtmek zorundadır. [2]

3.2.4 Uygun ve uygun olmayan dizayn (Feasible & Infeasible design)

Uygun dizayn, DVs ve SVs için belirtilen tüm sınırlamaları saęlayan dizayndır. Eęer sınırlamaların herhangi bir tanesi bile ihlal ediliyorsa dizayn uygun olmayan' dır. [2]

3.2.5 En iyi dizayn (best design)

Tüm sınırlamaları saęlamakla birlikte hedef fonksiyonunu minimum yapan dizayndır.

3.2.6 Analiz dosyası (Analysis file)

Birkaç değişik şekilde oluşturulabilen ve işlem öncesi (preprocessing), çözüm (solution) ve işlem sonrası (postprocessing) denilen tüm analiz safhalarını içeren bir ANSYS girdi dosyasıdır. Bu dosya DVs, SVs ve hedef fonksiyonu ifadelerini tanımlayan parametrelerle oluşturulmuş, yani parametrik olarak tanımlanmış bir model içermelidir. Bu dosya kullanılarak bir optimizasyon döngü dosyası, uzantısı ".loop" olmak üzere (örneğin; jobname.loop) otomatik olarak oluşturulur ve optimizasyon işleminde bu dosya kullanılır. Burada "jobname" (iş adı) yapılan çalışmaya verilen isimdir. [2]

3.2.7 Döngü (Loop)

Optimizasyon işleminde analiz dosyasının bir defa kullanılmasını ifade eder. Son döngü işleminden sonraki çıktılar, yani parametrelerin son değerleri "jobname.opo" adlı dosyada saklanır. [2]

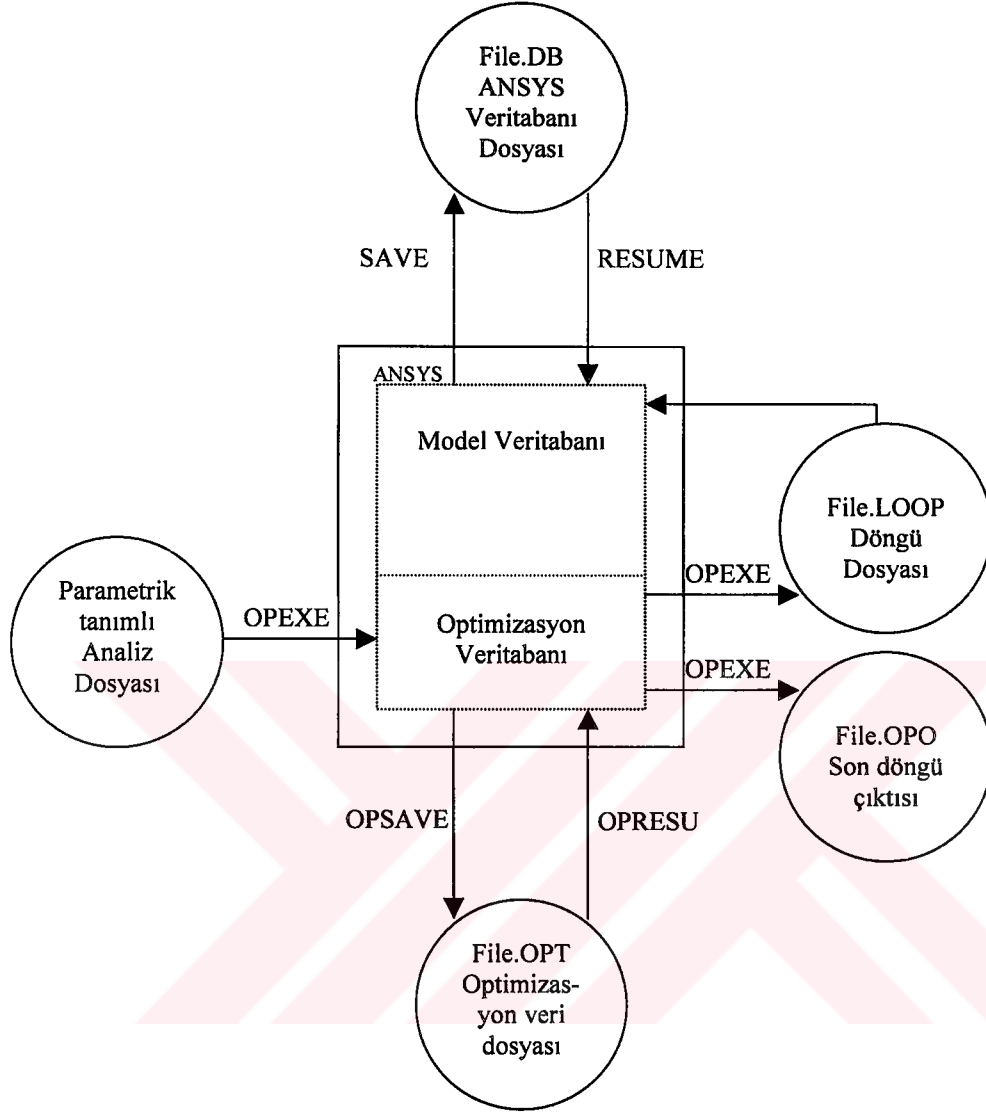
3.2.8 Optimizasyon iterasyonu (Optimization iteration)

Yeni bir dizayn setini oluşturan bir veya birden fazla döngüyü ifade eder. Birinci mertebe metodunda bir iterasyon bir döngüye eşittir. [2]

3.2.9 Optimizasyon veritabanı (Optimization database)

Optimizasyon veritabanı değişken tanımlamaları, parametreler ve tüm optimizasyon tanımlamalarını içerir. Bu veritabanı "jobname.opt" dosyasına kaydedilir.

Yukarıdaki kavramların bazıları bir şematik gösterim ile daha iyi anlaşılabilir. Şekil 3.1 bir optimizasyon işlemindeki bilgi akışını göstermektedir. [2]



Şekil 3.1- Optimizasyonda bilgi akışı [2]

3.3 Optimizasyon Değişkenlerinin Seçiminde Dikkat Edilecek Hususlar

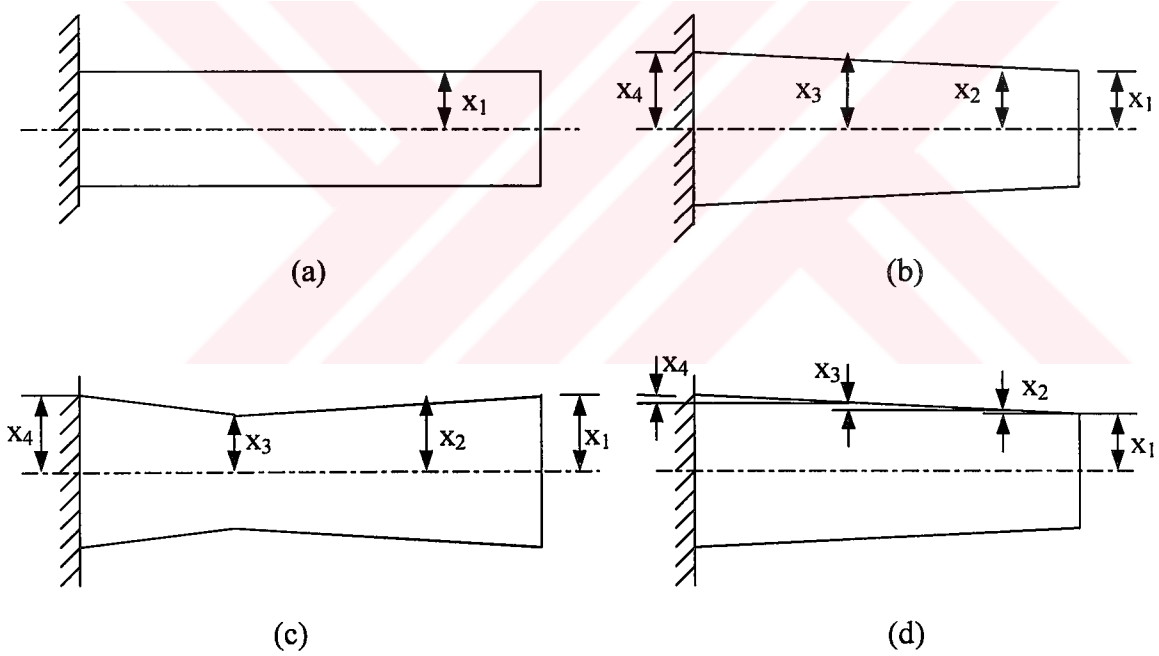
DVs, SVs ve hedef fonksiyonu seçiminde, göz önünde bulundurulduğunda fayda sağlayan birçok husus vardır. Bunlardan bazıları aşağıda zikredilmiştir.

3.3.1 Dizayn değişkenlerinin (DVs) seçimi

Dizayn değişkenleri genellikle uzunluk, kalınlık veya çap gibi geometrik büyüklüklerdir. Bunların muhakkak pozitif değerler alması gerekir. DVs seçimi yapılırken şunlara dikkat edilir:

- Mmkkn olduđu kadar az dizayn deęiřkeni kullanılmalıdır. Çok fazla dizayn deęiřkeni kullanmak doęru genel minimum noktaya yaklařmak yerine bir yerel minimum noktaya yaklařma ihtimalini artırır. Daha fazla dizayn deęiřkeni daha fazla iterasyonu ve buna baęlı olarak bilgisayarın iřlem sũresini artırır. Dizayn deęiřkeni sayısını azaltmanın bir yolu bazılarını, dięerleri cinsinden ifade etmektir. Fakat DVs gerçekten baęımsız deęiřkenler deęilse bu iřlem yapılamaz. Ancak optimum řeklin simetrik olacaęı dũřnũlmũřse, simetrik elemanlar iin bir DV kullanılabilir.

- OPVAR komutu ile DVs iin mantıklı sınırlamalar (min&max) verilir. Daha dar sınır verilirse daha iyi dizayn elde edilebilir.



řekil 3.2- Bir ankastre kiriř iin dizayn deęiřkenlerinin seilmesi [2]

- DVs optimum dizayna kolaylık saęlayacak řekilde seilir. rneęin; řekil 3.2.(a) da gsterilen bir ankastre kiriřin aęırlık optimizasyonunda sadece bir tane dizayn deęiřkeni (x_1) kullanılabilir. Ancak byle yapmak daha dũřk aęırlık verebilecek konik veya eęrisel yapıdaki bir dizayna izin vermeyecektir. Konik veya eęrisel bir dizayn iin řekil 3.2.(b) 'de ki gibi drt tane dizayn deęiřkeni (x_1, x_2, x_3, x_4) kullanılabilir. Fakat bu durumda da řekil 3.2.c 'de ki gibi pek de arzu edilmeyen bir

dizayn oluşabilecektir. Daha farklı bir yaklaşım da Şekil 3.2.(d) 'de ki gibi dizayn değişkenlerini birbirine göre izafi tanımlama şeklinde olabilir. Dizayn değişkenlerini seçerken gerçekçi olmayan veya arzulanmayan dizaynların oluşmasına meydan verilmemelidir. [2]

3.3.2 Dizayn sınırlamalarının (SVs) seçimi

Dizayn sınırlamalarına örnek olarak gerilmeler, sıcaklıklar, debiler, frekanslar, deformasyonlar, absorbe edilen enerji vb. verilebilir. Bunlara dizaynı sınırlayan değerler verilmektedir ve ANSYS tarafından hesaplanırlar. Seçimi yapılırken şu hususlar dikkate alınmalıdır:

- SVs tanımlanırken (OPVAR komutu ile) minimum değer belirtileceği alana bir şey yazılmazsa “alt limit yok” anlamına gelir. Aynı şekilde maksimum değer girileceği alan boş bırakılırsa “üst limit yok” anlamına gelir. Kullanılan ünlem (!) işaretleri açıklamaları belirtmektedir. Herhangi bir fonksiyonu yoktur.

OPVAR,SIG,SV,500,1000	! $500 \leq SIG \leq 1000$
OPVAR,SIG,SV,,1000	! $SIG \leq 1000$
OPVAR,SIG,SV,500,,	! $500 \leq SIG$

- Dizaynı en iyi şekilde sınırlandırmak için yeteri kadar SVs kullanılmalıdır.
- Altproblem yaklaşımı metodunda , eğer mümkünse DVs ile doğrusal veya ikinci dereceden (quadratic) ilişkisi olan SVs seçilir. Aksi taktirde iyi bir yaklaşım sağlanmayabilir.
- Eğer bir SV hem alt hem de üst sınırlamaya sahipse çok dar sınırlamalardan kaçınılır. Çünkü uygun dizayn olmayabilir. Örneğin 500 ile 1000 MPa arasındaki bir gerilme değeri sınırlaması, 900 ile 1000 MPa arasındaki bir sınırlamadan daha iyidir.

- Eğer bir eşitlik sınırlaması tanımlanacaksa (mesela frekans=386.4 Hz.), bu değeri ortasına alan iki tane SVs tanımlanır. Bu işlem aşağıdaki gibi yapılır. [2]

```
*GET,FREQ,ACTIVE,,SET,FREQ    ! Parametre FREQ=Hesaplanan frekans
FREQ1=FREQ
FREQ2=FREQ
....
/OPT
OPVAR,FREQ1,SV,,387            ! Üst limit FREQ1=387
OPVAR,FREQ2,SV,386,            ! Alt limit FREQ2=386
```

3.3.3 Hedef fonksiyonunun seçimi

Hedef fonksiyonu, minimum veya maksimum olmasını istediğimiz değerdir. Seçilirken şunlara dikkat edilir:

- ANSYS programı optimizasyon işlemini yaparken hedef fonksiyonunu daima minimize etmeye çalışır. Eğer problem gereği örneğin bir x değeri maksimum yapılmak isteniyorsa, x değeri cinsinden ifade edilen bir x_1 değeri minimum yapılacak şekilde problem formüle edilir. Bunu $x_1=C-x$ veya $x_1=1/x$ şeklinde iki yolla yapabiliriz. Burada C değeri, x için çıkması muhtemel olan değerlerin çok üzerindeki bir sayı olarak alınır. $C-x$ hedef fonksiyonunu tanımlamak için $1/x$ 'e göre daha iyi bir yoldur. Çünkü ikincisi ters orantılı, dolayısıyla doğrusal olmayan bir ifade olduğundan altproblem yaklaşımı metodunda kullanılan yaklaşımlarla doğru olarak temsil edilemez.
- Hedef fonksiyonu optimizasyon işlemi yapılırken pozitif kalmalıdır. Çünkü negatif değerler sayısal problemlere neden olabilir. Negatif değerlerin oluşmasından kaçınmak için, hedef fonksiyonunun beklenen en yüksek değerinden daha büyük bir sayı eklenebilir. [2]

3.4 ANSYS ile Dizayn Optimizasyonunun Prosedürü

Bir ANSYS optimizasyonu için temelde iki yol vardır:

1. Bir batch dosyası oluşturup bu dosyayı daha sonra çalıştırarak. Bu dosya, bir text editörü (Örneğin; Notepad, Wordpad) kullanılarak gerekli olan bütün komutlar sırası ile yazılmak sureti ile oluşturulur.
2. Grafik Kullanıcı Arayüzü (GUI: Graphical User Interface) ile işlemleri interaktif olarak gerçekleştirerek. Bu yöntemde ise ANSYS Software programı açılıp, menüler ve diyalog kutuları yardımı ile optimizasyon işlemi yapılır. Arzu edildiğinde menüleri kullanmaksızın girdi kutusu 'na (ANSYS Input) aynen batch dosyasının oluşturulması gibi, komutlar gerekli argümanları ile birlikte sırayla girilerek ve her birinin ardından Enter tuşuna basılarak da yapılabilir.

Eğer kullanıcı ANSYS komutlarına yeterince hakim ise, özellikle kompleks analizler için (örneğin; nonlinear) birinci yol daha uygun bir yaklaşım olabilir.

Dizayn optimizasyonu prosedürü aşağıdaki ana adımları içerir. Bu adımlar optimizasyonun GUI veya batch modu ile yapılması durumunda bazı küçük farklılıklar gösterebilmektedir.

1. Döngü için kullanılacak analiz dosyasının oluşturulması. Bu dosya aşağıdaki bölümleri içerir.
 - a. Modeli parametrik olarak oluşturmak (PREP7 modülü)
 - b. Çözümü yapmak (SOLU modülü)
 - c. SVs ve hedef fonksiyonu olarak kullanılan parametrelere cevap değerlerini atamak
2. Optimizasyon için parametrelerin tanıtılması
3. Optimizasyon modülüne geçiş ve analiz dosyasının belirtilmesi (OPT modülü)
4. Optimizasyon değişkenlerinin belirtilmesi (OPT)
5. Optimizasyon aracının veya metodunun seçilmesi (OPT)
6. Optimizasyon döngü kontrollerinin belirtilmesi (OPT)

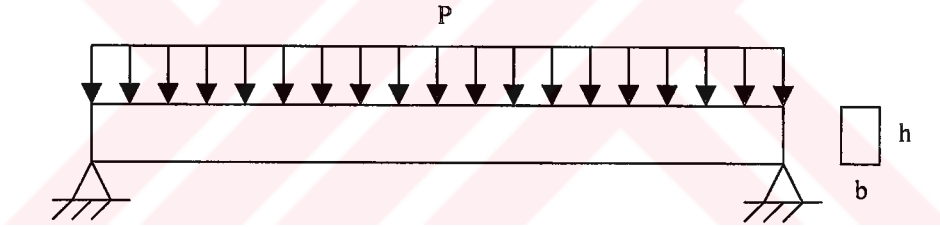
7. Optimizasyon analizinin başlatılması (OPT)

8. Dizayn setleri sonuçlarının (OPT) ve analiz sonuçlarının (POST1 ve POST26) incelenmesi [2]

Bu maddelerle ilgili detaylar aşağıda verilmiştir. Batch ve GUI yaklaşımları arasındaki farklar gerektiği yerlerde belirtilmiştir. Şimdi basit bir problemi ele alarak tüm bu işlem safhalarını daha iyi anlamaya çalışacağız.

Problem: Şekilde görülen kirişin ağırlığı belirtilen sınırlamalar dahilinde minimum yapılacaktır.

- Toplam σ gerilmesi $\sigma_{\max}$ değerini aşmayacak [$\sigma < \sigma_{\max}$]
- Kirişte oluşan δ sehim $\delta_{\max}$ değerini aşmayacak [$\delta < \delta_{\max}$]
- Kirişin h yüksekliği $h_{\max}$ değerini aşmayacak [$h < h_{\max}$]



Şekil 3.3- Dizayn optimizasyonu için bir örnek [2]

3.4.1 Adım 1- Analiz dosyasının oluşturulması

Analiz dosyası ANSYS optimizasyonunun çok önemli bir parçasıdır. Program bu dosyayı analiz döngülerinde kullanılan döngü dosyasını oluşturmak için kullanır. Herhangi bir tipteki ANSYS analizi (structural, thermal, magnetic v.b.; linear, nonlinear) analiz dosyasının bir parçası yapılabilir.

Bu dosyada model genellikle DVs olan parametrelerle tanımlanmalıdır. Sonuçlar ise SVs ve hedef fonksiyonunu ifade eden parametrelerle tanımlanmalıdır.

Dosya optimizasyon esnasında tekrar tekrar kullanılacağından dosyadaki gereksiz komutlar (görüntü, listeleme v.b. ile ilgili) temizlenmelidir. Aksi takdirde gereksiz işlemler her döngüde yapılarak verimi düşüreceklerdir.

Analiz dosyasını oluşturmanın iki yolu vardır:

1. Bir sistem editörü ile komutlar satır satır girilerek.
2. İnteraktif olarak ANSYS'le model oluşturulurken bilgisayar tarafından yazılan "command.log" dosyası analiz dosyası olarak kullanılır.

Dosyayı bir sistem editörü ile oluşturmak, batch dosyası oluşturmak ile aynıdır. Analiz dosyası, batch modundaki bütün bir optimizasyon dosyasının genellikle ilk bölümünü oluşturur. Bu metotla bütün optimizasyon dosyasının ve tabii ki analiz dosyasının kontrolü ve gerekli değişikliklerin yapılması mümkündür. Fakat kullanıcı ANSYS komutlarına ve bunların yapılarına hakim değilse bu yöntem uygun olmayabilir.

ANSYS ' le çalışmaya yeni başlayan bir kullanıcı için interaktif olarak model oluşturulup, model oluşturulurken ANSYS tarafından yazılan command.log dosyasını analiz dosyası olarak kullanmak daha kolay bir yöntemdir. Fakat bu safhada command.log dosyasında yazılmış bazı gereksiz komutların temizlenmesi gerekir. Bu işlem optimizasyon esnasında gereksiz tekrarlanmaları önler. [2]

3.4.1.1 Modeli parametrik olarak oluşturmak

DVs parametreleri kullanılarak modeli oluşturmak için PREP7 modülü kullanılır. Giriş örneğimiz için dizayn parametreleri B (genişlik) ve H (yükseklik) tır. Dolayısıyla gerçek sabitleri (real constants) bunlara göre ifade edeceğiz.

...

```
/PREP7                                ! PREPROCESSOR (Ön işlemci) 'e giriş
!      DV parametrelerine başlangıç değerlerinin atanması.
B=2.0                                ! Başlangıç genişliği
```

H=3.0	! Başlangıç yüksekliği
!	
ET,1,BEAM3	! 2-D kiriş elemanı
AREA=B*H	! Kirişin kesit alanı
IZZ=(B*(H**3))/12	! Z-eksenine göre atalet momenti
R,1,AREA,IZZ,H	! Gerçek sabitlerin belirtilmesi
!	
! Modelin geriye kalan kısmı	
MP,EX,1,30E6	! Malzemeye ait Young Modülü
N,1	! Node ' ların oluşturulması
N,11,120	
FILL	
E,1,2	! Elemanların oluşturulması
EGEN,10,1,-1	
FINISH	! PREP7 modülünden çıkış.
...	

Dizayn parametreleri genellikle /PREP7 de olmakla birlikte herhangi bir yerde tanımlanabilir. Bunlara atanan değerler bir başlangıç dizaynını oluşturur. [2]

3.4.1.2 Çözümü yapmak

SOLUTION işlemcisi analizin tipi, analiz seçenekleri, yüklerin uygulanması, sonlu elemanlar çözümünün (FEM: Finite Element Method) başlatılması gibi işlemlerin tanımlanması için kullanılır. Analiz için gerekli olan serbestlik dereceleri, doğrusal olmayan analiz için yaklaşım kriteri, harmonik cevap analizi için frekans sınırı, vb. özellikler belirtilir. Yükler ve sınır şartları da DVs olabilir.

Kiriş problemimiz için SOLUTION girdileri aşağıdaki gibi olacaktır.

...	
/SOLU	
ANTYPE,STATIC	! Analiz tipinin belirtilmesi (statik)

D,1,UX,0,,11,10,UY	! Kirişin iki ucundaki deformasyonlar
	UX=UY=0
SFBEAM,ALL,1,PRES,100	! Birim uzunluğa uygulanan basınç =100
SOLVE	
FINISH	! SOLUTION 'dan çıkış

Bu adım sadece bir analizle sınırlı değildir. Örneğin ısı bir çözüm ardından da gerilme çözümü yapılabilir (ısı gerilme hesaplamaları için). [2]

3.4.1.3 Sonuçların elde edilmesi

Bu adımda çözüm sonucu ulaşılan verilerin elde edilmesi ve bunların genellikle SVs ve hedef fonksiyonunu temsil eden parametrelere atanması sağlanır. ANSYS tarafından hesaplanan değerler *GET komutu yardımı ile parametrelere atanmaktadır (menü yolu → **Utility Menu > Parameters > Get Scalar Data**). Bu aşamada /POST1 kullanılacaktır.

Kiriş problemimizde kirişin ağırlığı hedef fonksiyonu (minimum yapılacak) olarak kullanılacaktır. Ağırlık hacimle doğru orantılı olduğundan, yoğunluğun homojen olduğunu kabul ederek, kirişin toplam hacminin minimum yapılması ağırlığının da minimum yapılması demektir. Bu yüzden hacmi hedef fonksiyonu olarak kullanacağız. Problemdaki SVs, toplam gerilme ve deformasyon 'dur. [2]

...	
/POST1	
SET,...	
NSORT,U,Y	! UY yerdeğiştirmesine göre node' ların sınıflandırılması
*GET,DMAX,SPORT,,MAX	! Parametre DMAX=Maksimum yerdeğiştirme
!	
ETABLE,VOLU,VOLU	! VOLU=Her bir elemanın hacmi

ETABLE,SMAX_I,NMISC,1	! SMAX_I=Her bir elemanın I ucundaki maksimum gerilme
ETABLE,SMAX_J,NMISC,3	! SMAX_J= Her bir elemanın J ucundaki maksimum gerilme
!	
SSUM	! Eleman tablosunun her bir kolonundaki dataların toplanması
*GET,VOLUME,SSUM,,ITEM,VOLU	! Parametre VOLUME=toplam hacim
ESORT,ETAB,SMAX_I,,1	! SMAX_I 'nın değerine göre elemanların sınıflandırılması
*GET,SMAXI,SORT,,MAX	! Parametre SMAXI= SMAX_I 'nın maksimum değeri
ESORT,ETAB,SMAX_J,,1	! SMAX_J 'nın değerine göre elemanların sınıflandırılması
*GET,SMAXJ,SORT,,MAX	! Parametre SMAXJ= SMAX_J 'nın maksimum değeri
SMAX=SMAXI>SMAXJ	! Parametre SMAX=maksimum gerilme (SMAXI ve SMAXJ 'den büyük olanı)
FINISH	
...	

3.4.1.4 Analiz dosyasının hazırlanması

Buraya kadar analiz dosyasının oluşturulması için gerekli işlemler tanımlandı. Eğer bu dosya batch modunda hazırlanmışsa dosya basitçe kaydedilip ikinci adıma geçilebilir. Fakat eğer dosya interaktif olarak ANSYS ile hazırlanmışsa analiz dosyasını elde etmek için bazı düzenlemeler yapmak gerekir. Bu ise iki yolla yapılabilir:

1. Database Command Log: LGWRITE komutu kullanılarak (menü yolu → **Utility Menu >File >Write DB Log File**) model veritabanını oluşturan bir komut log dosyası oluşturulabilir. LGWRITE komutu Jobname.LGW adlı bir dosya oluşturur. Aktif veritabanını oluşturmak için kullanılmış tüm komutları içerir.

2. Session Log File: Jobname.LOG bir interaktif oturum esnasında kullanılmış olan tüm komutları içerir. Bu dosyayı analiz dosyası olarak kullanmak için bir sistem editörünü kullanarak gereksiz olan tüm komutlarının silinmesi gerekir. Eğer analiz birkaç oturumda gerçekleştirilmişse bütün halinde bir analiz dosyası oluşturmak için log dosyası parçalarını bir araya getirmek gerekir. [2]

3.4.2 Adım 2- Optimizasyon için parametrelerin tanıtılması

Analiz dosyasının tamamlanması ile birlikte artık optimizasyona hazır hale gelinir. İnteraktif olarak çalışıldığında veritabanındaki parametrelerin tanıtılması için aşağıdaki yollardan birisi kullanılır:

- Analiz dosyası ile birleştirilmiş veritabanı dosyası (Jobname.DB) çağırılır. Bu işlem, parametreleri de içeren bütün model veritabanını ANSYS 'e tanıtır. Veritabanı dosyasını çağırmak için aşağıdaki yollardan birisi kullanılır:

Komut: RESUME

GUI: Utility Menu >File >Resume Jobname.db
Utility Menu >File >Resume from

- Analiz dosyasının ANSYS 'e okutulması. Bu tüm model veritabanını ANSYS 'e tanıtır, fakat geniş bir model için zaman israfı olabilir. Analiz dosyasını okutmak için aşağıdaki yollardan birisi kullanılır:

Komut: /INPUT

GUI: Utility Menu >File >Read input from

- PARSAV komutu veya Utility Menu >Parameters >Save Parameters kullanılarak daha önceden oluşturulmuş bir parametre dosyasının çağırılması. Parametreleri çağırmak için aşağıdaki iki yoldan birisi kullanılabilir:

Komut: PARRES

GUI: Utility Menu >Parameters >Restore Parameters

- Analiz dosyasında oluşturulmuş parametrelerin tekrar oluşturulması. Bunu yapmak için analiz dosyasında hangi parametrelerin tanımlandığını bilmek gerekir. Aşağıdaki yollardan birisi kullanılır.

Komut: *SET veya “=” komutu

GUI: Utility Menu >Parameters >Scalar Parameters

Optimizasyon değişkenlerini tanımlamak için yukarıda zikredilen yollardan hiçbirini seçmeyip yerine Bölüm 3.4.4 ‘de anlatılan yöntem kullanılabilir. [2]

3.4.3 Adım 3- Optimizasyon modülüne giriş ve bunda kullanılacak analiz dosyasının belirtilmesi

Bu aşamadan itibaren tüm adımlar OPT işlemcisi içerisinde yapılacaktır. Optimizer ‘a (optimizasyon modülü) ilk girildiğinde ANSYS veritabanında var olan tüm parametreler otomatik olarak “dizayn set numarası 1” (design set number 1) olarak alınır. Optimizer ‘a giriş için aşağıdaki yollardan birisi kullanılır:

Komut: /OPT

GUI: Main Menu >Design Opt

İnteraktif modda analiz dosyasının ismini belirtmek gerekir. Analiz dosyası, optimizasyon döngü dosyasını (Jobname.LOOP) oluşturmak için kullanılacaktır. Default olarak herhangi bir analiz dosyası seçilmiş değildir. Bu yüzden analiz dosyası muhakkak belirtilmelidir. Bunun için aşağıdaki yollardan birisi kullanılır:

Komut: OPANL

GUI: Main Menu >Design Opt >Assign

Batch modundaki bir optimizasyon işlemi için, analiz dosyası, genellikle batch dosyasının ilk bölümünü oluşturur. Bu bölüm ilk satırdan itibaren /OPT ‘ nin ilk görüldüğü yere kadardır. Batch modunda analiz dosyası, batch dosyasından kopyalanan Jobname.BAT dosyasına default haldedir. Bu yüzden normalde analiz

dosyasının belirtilmesine gerek yoktur. Ancak herhangi bir sebepten dolayı batch optimizasyon girdisi iki dosyaya ayrılmışsa (birisi analiz dosyasını, diğeri ise tüm optimizasyon işlemlerini temsil eden) Optimizer 'a [/OPT] girdikten sonra analiz dosyasının belirtilmesi gerekir [OPANL]. [2]

3.4.4 Adım 4- Optimizasyon değişkenlerinin belirtilmesi

Bu adımda hangi parametrelerin DVs, hangilerinin SVs ve hangisinin hedef fonksiyonu olduğu belirtilir. Daha önce de belirttiğimiz gibi 60 taneye kadar DVs, 100 taneye kadar SVs, ve bir tane de hedef fonksiyonu tanımlanabilir. Optimizasyon değişkenlerinin belirtilmesi için şu yollardan birisi takip edilir:

Komut: OPVAR

GUI: Main Menu >Design Opt >Design Variables

Main Menu >Design Opt >State Variables

Main Menu >Design Opt >Objective

DVs ve SVs için minimum ve maksimum sınırlar belirtilir, hedef fonksiyonu için sınırlama gerekmemektedir. Ayrıca her bir değişken için bir tolerans belirtilir.

Eğer OPVAR komutu ile belirtilen parametre ismi daha önceden var olan bir parametre değilse, bu parametre ANSYS veritabanına otomatik olarak tanıtılır ve sayısal değer olarak sıfır atanır.

Daha önce tanımlanmış bir optimizasyon değişkeni istenildiği zaman yeniden tanımlanarak değiştirilebilir.

Var olan bir optimizasyon değişkeni silinebilir [OPVAR,İsim,DEL]. Menüdeki “delete” seçeneği parametreyi silmez, sadece optimizasyon değişkeni olarak pasif hale getirir. [2]

3.4.5 Adım 5- Optimizasyon aracı veya metodunun seçilmesi

ANSYS programında birkaç değişik optimizasyon aracı ve metodu kullanılabilir. Haklarında daha geniş bilginin Bölüm 5 'de verildiği bu metot ve araçlar şunlardır:

Optimizasyon metotları:

- Alt problem yaklaşımı metodu (Subproblem Approximation Method)
- Birinci mertebe metodu (First-order Method)
- Kullanıcı tanımlı metot (User Defined Method)

Optimizasyon araçları:

- Tek döngü çalıştırma (Single Loop Run)
- Rasgele Dizayn oluşturma (Random Design Generation)
- Süpürme (Sweep Generation)
- Faktöriyel Değerlendirme (Factorial Evaluation)
- Eğim Değerlendirme (Gradient Evaluation)

Optimizasyon aracı olarak “Single loop” default durumdadır. Bir araç veya metot belirtmek için aşağıdaki yollardan birisi kullanılır:

Komut: OPTYPE

GUI: Main Menu >Design Opt >Method/Tool

3.4.6 Adım 6- Optimizasyon döngü kontrollerinin belirtilmesi

Her bir metot ve araç maksimum iterasyon sayısı vb. kendine has döngü kontrollerine sahiptir. Bu kontrolleri ayarlamak için kullanılan tüm komutlara aynı menü yolundan ulaşılır:

GUI: Main Menu >Design Opt >Method/Tool

Kontrolleri ayarlayan komutlar şunlardır:

- Kontrolleri altproblem yaklaşımı metoduna ayarlamak için:

Komut: OPSUBP

ve

Komut: OPEQN

- Kontrolleri birinci merteye metoduna ayarlamak için:

Komut: OPFRST

- Kontrolleri rasgele dizayn oluşturma aracına ayarlamak için:

Komut: OPRAND

- Kontrolleri süpürme aracına ayarlamak için:

Komut: OPSWEEP

- Kontrolleri faktöriyel değerlendirme aracına ayarlamak için:

Komut: OPFACT

- Kontrolleri eğim değerlendirme aracına ayarlamak için:

Komut: OPGRAD

- Kontrolleri kullanıcı optimizasyon aracına ayarlamak için:

Komut: OPUSER

Ayrıca optimizasyon esnasında elde edilen verilerin nasıl saklanacağına dair bazı genel kontroller de şunlardır:

- Optimizasyon verilerinin saklanacağı dosyanın belirtilmesi (default olarak Jobname.OPT). aşağıdaki yollardan birisi kullanılabilir:

Komut: OPDATA

GUI: Main Menu >Design Opt >Controls

- Detaylı bir özeti alabilmek için aşağıdaki yollardan birisi kullanılabilir:

Komut: OPPRNT

GUI: Main Menu >Design Opt >Controls

- En iyi dizayn setinin saklanıp saklanmayacağını belirtmek için aşağıdaki yollardan birisi kullanılabilir (default olarak veritabanı ve sonuç dosyalarına sadece en son dizayn seti kaydedilir).

Komut: OPKEEP

GUI: Main Menu >Design Opt >Controls

Ayrıca döngü esnasında analiz dosyasının nasıl okunacağını da içeren çeşitli döngü karakteristikleri de kontrol edilebilir. Dosya ilk satırdan (default) veya /PREP7 'nin ilk bulunduğu yerden itibaren okutulabilir. Dizayn değişkenleri olarak belirtilen parametreler göz ardı edilebilir (default) veya döngü esnasında işleme sokulabilir. Bunlara ek olarak, hangi parametrelerin optimizasyon esnasında saklanacağı (sadece skalar parametreler (default) veya skalar ve array parametreler) belirtilebilir [2]. Bu döngü karakteristiklerini kontrol etmek için aşağıdaki yollardan birisi kullanılabilir.

Komut: OPLOOP

GUI: Main Menu >Design Opt >Controls

3.4.7 Adım 7- Optimizasyon analizinin başlatılması

Tüm uygun olan kontroller belirtildikten sonra, döngü artık başlatılabilir. Bunun için aşağıdaki yollardan birisi kullanılır:

Komut: OPEXE

GUI: Main Menu >Design Opt >Run

Optimizasyonun başlatılması ile birlikte analiz dosyası kullanılarak bir optimizasyon döngü dosyası (Jobname.LOOP) yazılır. Bu dosya analiz döngülerinde kullanılır. Döngü işlemi aşağıdaki durumlar oluşana kadar devam edecektir.

- *Yaklaşım:* Belirtilen yaklaşımın sağlanması
- *Durdurma:* Yaklaşım sağlanmadığı halde, maksimum döngü limitine ulaşılması veya art arda maksimum sayıda uygun olmayan sonuçlar elde edilmesi.
- *Tamamlanma:* Örneğin rasgele dizayn oluşturma için belirtilen döngü sayısına ulaşılması.

Eğer bir döngü, model ile ilgili bir sorundan dolayı kesilirse (örneğin; meshing hatası, yaklaşım sağlanamayan bir doğrusal olmayan çözüm, dizayn değişkenlerinin uyuşmazlığı vb.) optimizasyon o döngüyü bırakır fakat döngü işlemlerine devam eder. Eğer interaktif modda çalışılıyorsa bu durum bir uyarı ile bildirilir ve kullanıcıya devam edip etmemek istediği sorulur. Batch modunda ise döngü işlemi otomatik olarak devam eder.

Optimizasyon değişkenlerinin tüm değerleri ve diğer parametreler her iterasyonun sonunda optimizasyon veri dosyasında (Jobname.OPT) saklanır. 130 taneye kadar optimizasyon değişken seti saklanabilir. Bu sayıyı aşan her dizayn seti oluşturduğunda o andaki en kötü dizayn seti kayıttan silinir. [2]

Adım3 'ten buraya kadar açıklanan tüm sayfalar batch modunda aşağıdaki gibi olacaktır:

...

/OPT

! Optimizasyona giriş

OPANL,.....

! Analiz dosyasının isminin belirtilmesi (batch modu için gerekli değil)

!

! Optimizasyon değişkenlerinin belirtilmesi

OPVAR,B,DV,,5,16.5

! DVs → B ve H parametreleri

OPVAR,H,DV,,5,8

OPVAR,DMAX,SV,-0.1,0

! SVs → DMAX ve SMAX parametreleri

OPVAR,SMAX,SV,0,20000

OPVAR,VOLUME,OBJ

! Hedef fonksiyonu → VOLUME
parametresi

!

! Optimizasyon tipi ve kontrollerin belirtilmesi

OPTYPE,SUBP

! Altproblem yaklaşımı metodu

OPSUBP,30

! Maksimum iterasyon sayısı

OPEXE

! Optimizasyon döngüsünün başlatılması

3.4.8 Adım 8- Dizayn setleri sonuçlarının incelenmesi

Optimizasyon döngü işlemi tamamlandıktan sonra oluşan dizayn setleri, komutlarla veya menü yolları kullanılarak değişik yöntemlerle incelenebilir.

Belirtilen set numaralarının parametre değerlerini listelemek için aşağıdaki yollardan birisi kullanılır:

Komut: OPLIST

GUI: Main Menu >Design Opt >List

Tüm skalar parametreler veya sadece optimizasyon değişkeni olarak kullanılanlar listelenebilir.

Set numarasına göre bir değişkenin iterasyondan iterasyona nasıl değiştiği grafiklerle gözlemlenebilir. Bunun için aşağıdaki yollardan birisi kullanılır:

Komut: PLVAROPT

GUI: Main Menu >Design Opt >Graphs/Tables

Grafiğin apsisini set numarasından herhangi bir parametreye değiştirmek için aşağıdaki yollardan birisi kullanılır:

Komut: XVAROPT

GUI: Main Menu >Design Opt >Graphs/Tables

Belirtilen parametrelerin değişimlerini tablo halinde elde etmek de mümkündür.

Bunun için aşağıdaki iki yoldan birisi kullanılır:

Komut: PRVAROPT

GUI: Main Menu >Design Opt >Graphs/Tables

Süpürme, faktöriyel ve eğim araçlarının neticelerini listelemek için aşağıdaki yollar kullanılır.

Süpürme aracı için:

Komut: OPRSW

GUI: Main Menu >Design Opt >Tool Results > Print

Faktöriyel aracı için:

Komut: OPRFA

GUI: Main Menu >Design Opt >Tool Results > Print

Eğim aracı için:

Komut: OPRGR

GUI: Main Menu >Design Opt > Print

Yine bu araçların sonuçlarını, aşağıdaki yollarla grafik olarak çizdirmek de mümkündür:

Süpürme aracı için:

Komut: OPLSW

GUI: Main Menu >Design Opt > Tool Results > Sweeps

Faktöriyel aracı için:

Komut: OPLFA

GUI: Main Menu >Design Opt > Tool Results > Factorial

Eğim aracı için:

Komut: OPLGR

GUI: Main Menu >Design Opt > Tool Results > Gradient

Optimizasyon hakkında bilgi edinmenin bir başka yolu da STATUS komutunu kullanmaktır (menü yolu → **Main Menu >Design Opt >Status**). Bu komut analiz dosyasının ismini, kullanılacak optimizasyon tekniğini, var olan dizayn setlerini, optimizasyon değişkenlerini vb. liste halinde verir. Bu komutu kullanmak arzulanan ayarların girilip girilmediğini kontrol etmek için çok faydalıdır. Optimizasyon verilerinin elde edilmesine yönelik tüm bu işlemlerin yanında, analiz sonuçlarını da POST1 ve POST26 modüllerinden elde etmek mümkündür. Default olarak, son dizayn seti için sonuçlar Jobname.RST dosyasına kaydedilir (burada dosya uzantısının son iki harfi analizin tipini gösterir. Örneğin Jobname.RST → statik analiz, Jobname.RTH → Termal analiz vb.). En iyi dizayn için oluşan veritabanı ve dizayn setini kaydetmek için döngü başlamadan önce aşağıdaki yollarda birisi uygulanır:

Komut: OPKEEP,ON

GUI: Main Menu >Design Opt >Controls

Bu işlemle “en iyi sonuçlar” Jobname.BRST (.BRTH vb.) ve “en iyi veritabanı” Jobname.BDB dosyalarına kaydedilir. [2]

3.4.9 Dizayn setlerinin manipüle edilmesi

Dizayn setleri incelendikten sonra kullanıcının isteğine göre bazı düzenlemeler yapılabilir. Örneğin; bir rasgele dizayn oluşturma işleminden sonra uygun dizayn setlerini bir sonraki optimizasyonda veri noktaları olarak kullanmak için saklayıp, diğer bütün uygun olmayan dizaynlar kayıttan silinmek istenebilir. İstenmeyen dizayn setlerini kayıttan atmak için iki komut kullanılabilir.

- En iyi dizayn setlerinden belirtilen n adedini veya tüm uygun setleri seçmek için aşağıdaki yollardan birisi kullanılabilir:

Komut: OPSEL

GUI: Main Menu >Design Opt >Select/Delete

OPSEL komutu ile seçilmeyen tüm dizayn setleri optimizasyon veritabanından silinir.

- Verilecek aralıktaki dizayn setleri silinebilir. Bunu yapmak için dizayn seti numaralarına göre bir alt bir de üst sınır verilir. Aşağıdaki yollardan birisi kullanılır:

Komut: OPDEL

GUI: Main Menu >Design Opt >Select/Delete

Dizayn setleri ile ilgili diğer bazı komutlar da şunlardır:

- Varolan iki dizayn setini ekleyerek yeni bir dizayn seti oluşturmak için aşağıdaki yollardan birisi kullanılır.

Komut: OPADD

GUI: Main Menu >Design Opt >Combine

- Aktif skalar parametre değerlerini kullanarak yeni bir dizayn seti oluşturmak için aşağıdaki yollardan birisi kullanılır: [2]

Komut: OPMAKE
GUI: Main Menu >Design Opt >Create

3.4.10 Optimizasyonun birden fazla sayıda çalıştırılması

Optimizasyon işlemi birden fazla sayıda uygulanmak istenebilir. Bunun değişik nedenleri vardır. Örneğin yapılan optimizasyon istenen optimum noktayı bulamayabilir veya kullanıcı bir dizayn aracını kullanarak bilgi edinip bunu takiben optimizasyonu çalıştırmak isteyebilir. Yapılan ilk birkaç optimizasyon işleminden elde edilen bilgilere göre dizayn uzayı değiştirilip tekrar çalıştırılabilir. [2]

3.4.10.1 Aynı oturumda birden fazla optimizasyonun yapılması

Eğer yapılan optimizasyonların tümü aynı oturumda gerçekleştiriliyorsa prosedür oldukça basittir. Yapılan bir optimizasyon uygulamasının ardından tüm optimizasyon girdileri basitçe tekrar tanımlanır ve sonraki optimizasyon uygulaması çalıştırılır. Bunun için aşağıdaki yollardan birisi kullanılır:

Komut: OPEXE
GUI: Main Menu >Design Opt >Run

Eğer optimizasyon işlemi yapıldıktan sonra ANSYS programından çıkılıp başka bir zaman devam etmek istenirse aşağıdaki yol kullanılır. [2]

3.4.10.2 Farklı oturumlarda birden fazla optimizasyonun yapılması

Bir optimizasyon analizinin yeniden başlatılması için veritabanı dosyası (Jobname.OPT) çağrılır. Böyle yapmak için aşağıdaki yollardan birisi kullanılır:

Komut: OPRESU
GUI: Main Menu >Design Opt >Resume

Veri bir kere okunduktan sonra optimizasyon tipi, kontrolleri vb. yeniden belirtilip döngü işlemi başlatılabilir. Optimizasyonu çalıştırabilmek için çağrılan veritabanı ile uyumlu olan analiz dosyasının da bulunması gerekir. Analizi başlatmak için aşağıdaki yollardan birisi kullanılır:

Komut: OPEXE

Main Menu >Design Opt >Run

Tipik bir yeniden başlatma işlemi batch dosyasında şöyle görünür:

....

....

/OPT

OPRESU,....

! Optimizasyon dosyasının ismi
(Jobname.OPT)

OPSEL,10

! On tane en iyi dizaynı seç

OPTYPE,....

! Optimizasyon aracı veya metodunun
belirtilmesi

....

....

! Diğer optimizasyon girdilerinin yapılması

OPEXE

! Optimizasyonun başlatılması

Yeni yapılan optimizasyon işleminden sonra oluşan veritabanı çağırılan dosyanın üzerine yazılır.

Ayrıca batch modunda hazırlanan optimizasyon verisi interaktif oturumda yine aynı komutla çağırılabilir.

Eğer optimizasyon veritabanında bilgi varsa, çağırma işleminden önce bu bilgilerin silinmesi daha iyi olacaktır. Bu yapıldığı zaman tüm ayarlar default değerlerine dönüşecektir. Optimizasyon veritabanını silmek için aşağıdaki yollardan birisi kullanılır:

Komut: OPCLR

GUI: Main Menu >Design Opt >Clear&Reset

ANSYS veritabanı OPCLR komutundan etkilenmemektedir. Eğer çağırılan optimizasyon problemi bir öncekinden tamamen bağımsız olacaksa ANSYS veritabanını silmek gerekebilir [2]. ANSYS veritabanını temizlemek için aşağıdaki yollardan birisi kullanılır:

Komut: /CLEAR

GUI: Main Menu >Design Opt >Clear&Start New



BÖLÜM 4

ANSYS İLE OPTİMİZASYON UYGULAMALARI

4.1 Uygulama-1

Bu uygulamada birinci merteye metodu kullanılarak bir optimizasyon analizi yapılacaktır.

Problem tanımı:

Üç çubuklu bir destek sistemine (Şekil 4.1) yatay ve dikey kuvvetler etki etmektedir. Desteğin ağırlığı, maksimum gerilme değeri 400 psi ($2,76 \cdot 10^6$ Pa) değerini aşmamak şartı ile minimize edilecektir. Üç adet elemanın kesit alanlarının ve B uzunluğunun belirli sınırlar dahilinde değiştirilmesine izin verilmiştir. [2]

Analiz için aşağıdaki özelliklere sahip malzeme kullanılacaktır:

$$E=2.16E6 \text{ psi } (=1,449 \cdot 10^{10} \text{ Pa})$$

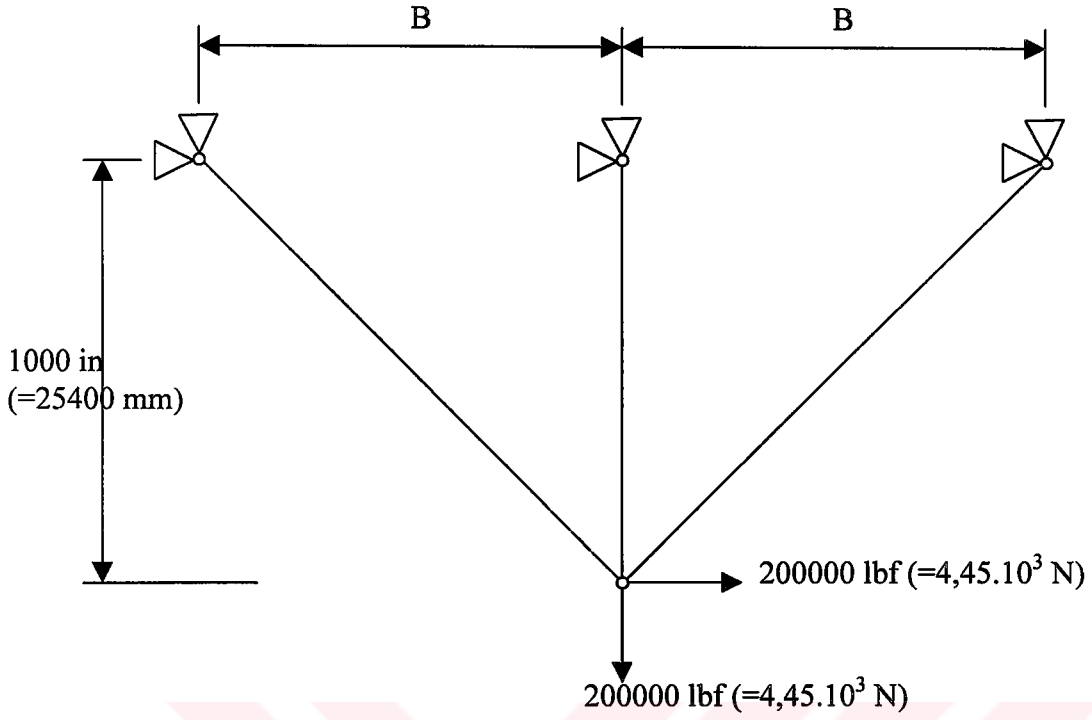
$$\rho=2.85E-4 \text{ lb/in}^3 (=7,896 \text{ kg/m}^3) \text{ (yoğunluk)}$$

$$\sigma_{\max}=400 \text{ psi } (2,76 \cdot 10^6 \text{ Pa})$$

Analiz için aşağıdaki geometrik özellikler kullanılacaktır:

$$1 \text{ in } (=25,4 \text{ mm}) \geq \text{Desteğin kesit alanı} \geq 1000 \text{ in}^2 (=25400 \text{ mm}^2)$$

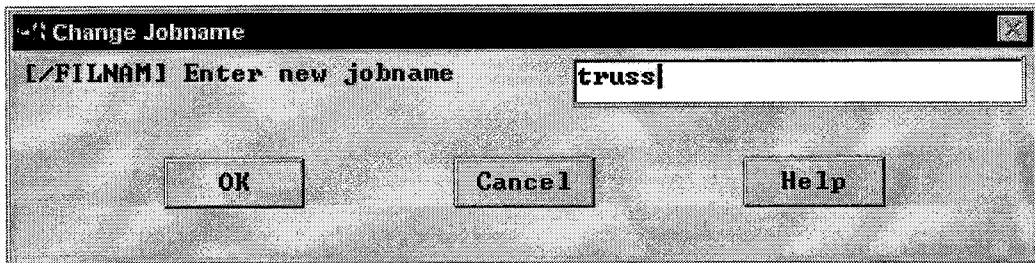
$$400 \text{ in } (=10160 \text{ mm}) \geq B \geq 1000 \text{ in } (=25400 \text{ mm})$$



Şekil 4.1- Problemin şematik gösterilişi [2]

4.1.1 Adım 1- İş adının (Jobname) belirtilmesi

1. İş adının belirtileceği diyalog kutusunu (Şekil 4.2) açmak için şu yol izlenir:
Utility Menu >File >Change Jobname



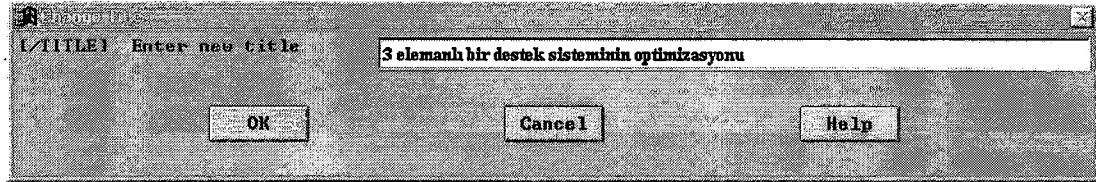
Şekil 4.2- İş adı diyalog kutusu

2. Jobname olarak "truss" yazılır.
3. Diyalog kutusunu kapatmak için "OK" butonuna basılır.

4.1.2 Adım 2- Analiz başlığının verilmesi

1. “Başlık değiştirme” diyalog kutusunu (Şekil 4.3) açmak için şu yol izlenir.

Utility Menu >File >Change Title



Şekil 4.3- Analiz başlığı diyalog kutusu

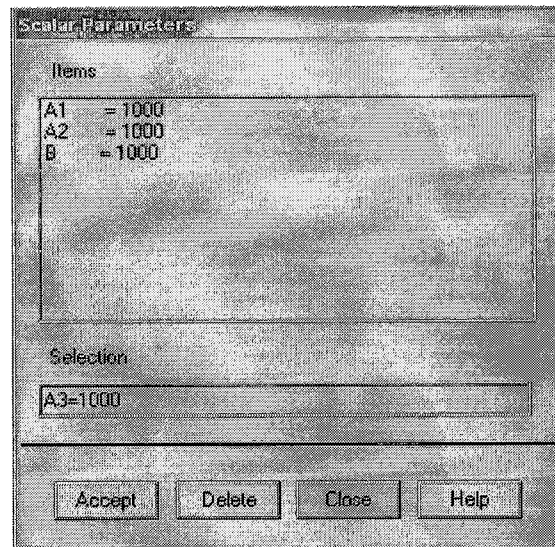
2. Yeni başlık olarak “3 elemanlı bir destek sisteminin optimizasyonu” yazılır.
3. Diyalog kutusu “OK” butonuna basılarak kapatılır.

4.1.3 Adım 3- Parametrelere başlangıç değerlerinin atanması

1. “Skalar Parametreler” diyalog kutusunu (Şekil 4.4) açmak için şu yol izlenir:

Utility Menu >File >Scalar Parameters

2. Aşağıdaki parametreler, “selection” kutusuna her birinin ardından “Enter” tuşuna veya “Accept” butonuna basılarak girilir.



Şekil 4.4- Parametrelere başlangıç değerlerinin atanması

B=1000

A1=1000

A2=1000

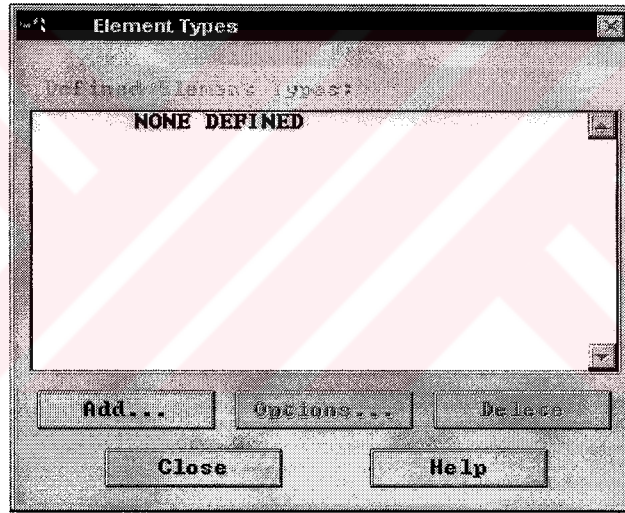
A3=1000

3. Diyalog kutusunu kapatmak için “Close” butonuna basılır.

4.1.4 Adım 4- Eleman tipinin tanımlanması

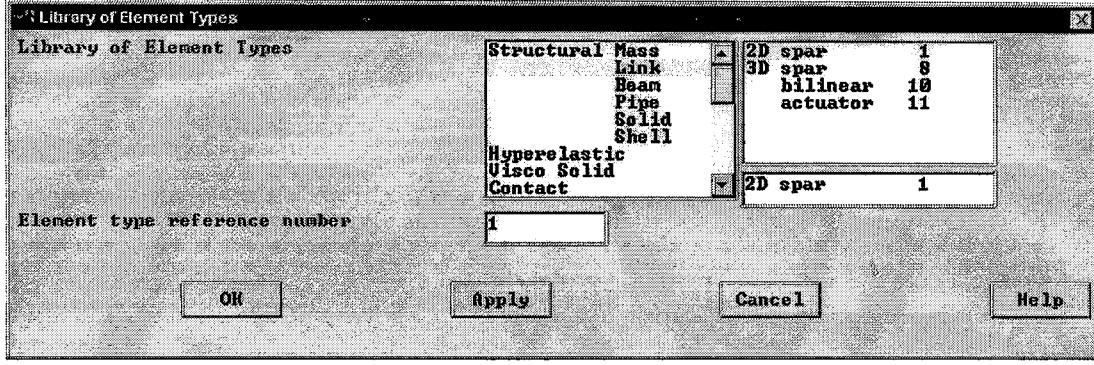
1. “Eleman tipleri” diyalog kutusunu (Şekil 4.5) açmak için şu yol izlenir.

Main Menu >Preprocessor >Element Type >Add/Edit/Delete



Şekil 4.5- Eleman tipleri diyalog kutusu

2. “Eleman tipleri kütüphanesini” (Şekil 4.6) açmak için “Add” butonuna basılır.
3. Sol sütundaki “Structural Link” seçilir.
4. Sağ kolondaki “2D Spar 1 (LINK1)” seçilir.
5. Eleman tipi referans numarası için 1 yazılır.
6. Eleman tipleri kütüphanesini kapatmak için “OK” butonuna basılır.
7. Eleman tipleri diyalog kutusunu kapatmak için “Close” butonuna basılır.

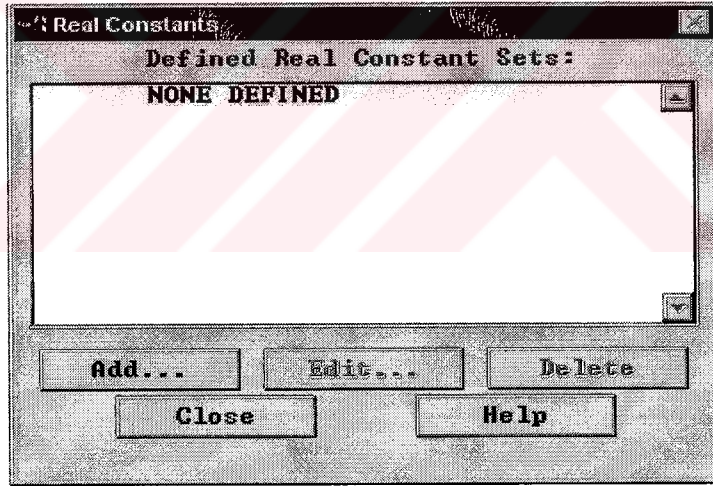


Şekil 4.6- Eleman tipleri kütüphanesi

4.1.5 Adım 5- Gerçek sabitlerin (Real constants) tanımlanması

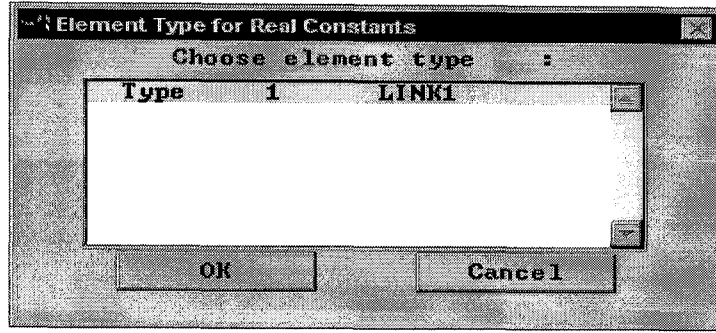
1. “Gerçek sabitler” diyalog kutusunu (Şekil 4.7) açmak için şu yol izlenir.

Main Menu >Preprocessor> Real Constants

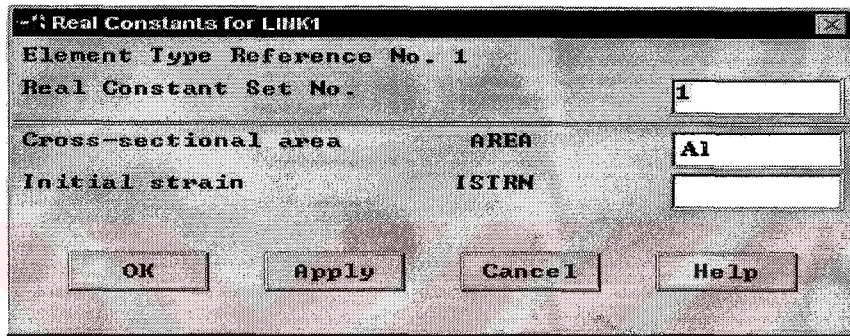


Şekil 4.7- Gerçek sabitler diyalog kutusu

2. “Gerçek sabitler için eleman tipi” diyalog kutusunu (Şekil 4.8) açmak için “Add” butonuna basılır.
3. “LINK1 eleman tipi için gerçek sabitler” diyalog kutusunu (Şekil 4.9) açmak üzere “OK” butonuna basılır.
4. Gerçek sabitler set numarası için 1 yazılır.
5. Kesit alanı için A1 yazılır.



Şekil 4.8- Gerçek sabitler için eleman tipi diyalog kutusu



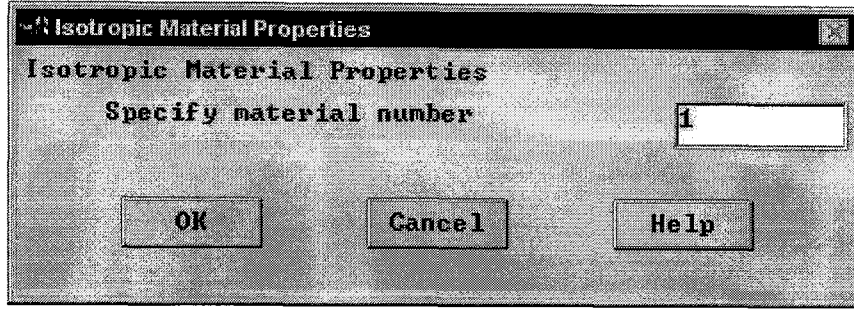
Şekil 4.9- LINK1 elemanı için gerçek sabitler diyalog kutusu

6. “Apply” butonuna basılır. Bu işlemle Set-1 için kesit alanına 1000 değeri yazılacaktır.
7. Gerçek sabitler set numarası için 2 yazılır.
8. Kesit alanı için A2 yazılır
9. “Apply” butonuna basılır. Bu işlemle Set-2 için kesit alanına 1000 değeri yazılacaktır
10. Gerçek sabitler set numarası için 3 yazılır.
11. Kesit alanı için A3 yazılır.
12. “OK” butonuna basılır.
13. Gerçek sabitler diyalog kutusunu kapatmak için “Close” butonuna basılır.

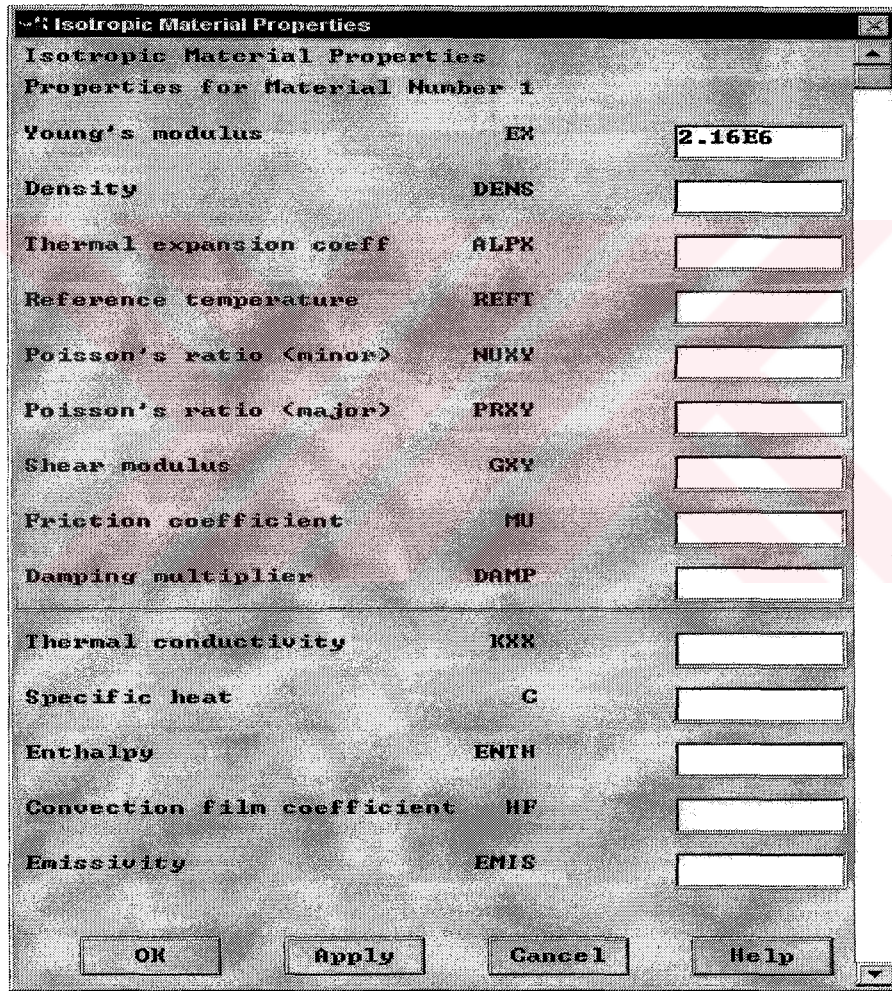
4.1.6 Adım 6- Malzeme özelliklerinin tanımlanması

1. “İzotropik malzeme özellikleri” diyalog kutusunu (Şekil 4.10) açmak için şu yol izlenir.

Main Menu >Preprocessor >Material Props >-Constant-Isotropic



Şekil 4.10- İzotropik malzeme özellikleri diyalog kutusu



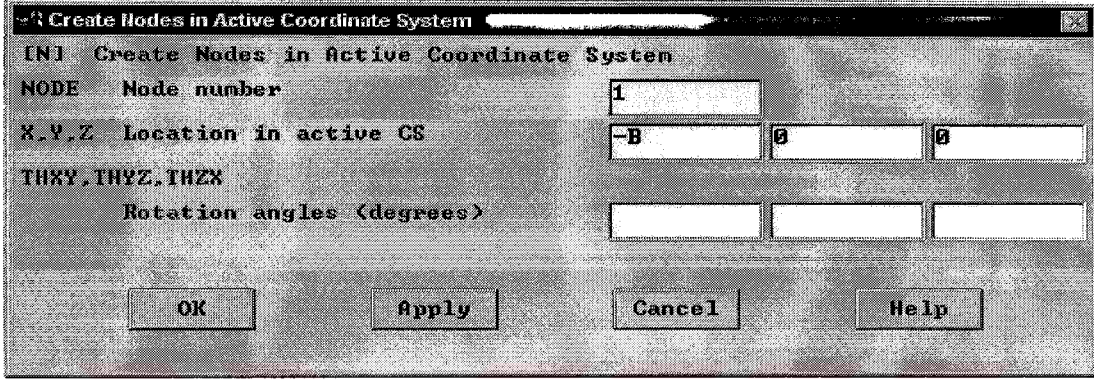
Şekil 4.11- Malzeme özelliklerinin belirtilmesi

2. Malzeme numarası için 1 yazılır.
3. "OK" butonuna basılır.
4. Yeni diyalog kutusuna Young modülü (EX) için 2.16E6 yazılır.(Şekil 4.11)
5. "OK" butonuna basılır.

4.1.7 Adım 7- Node 'ların oluşturulması

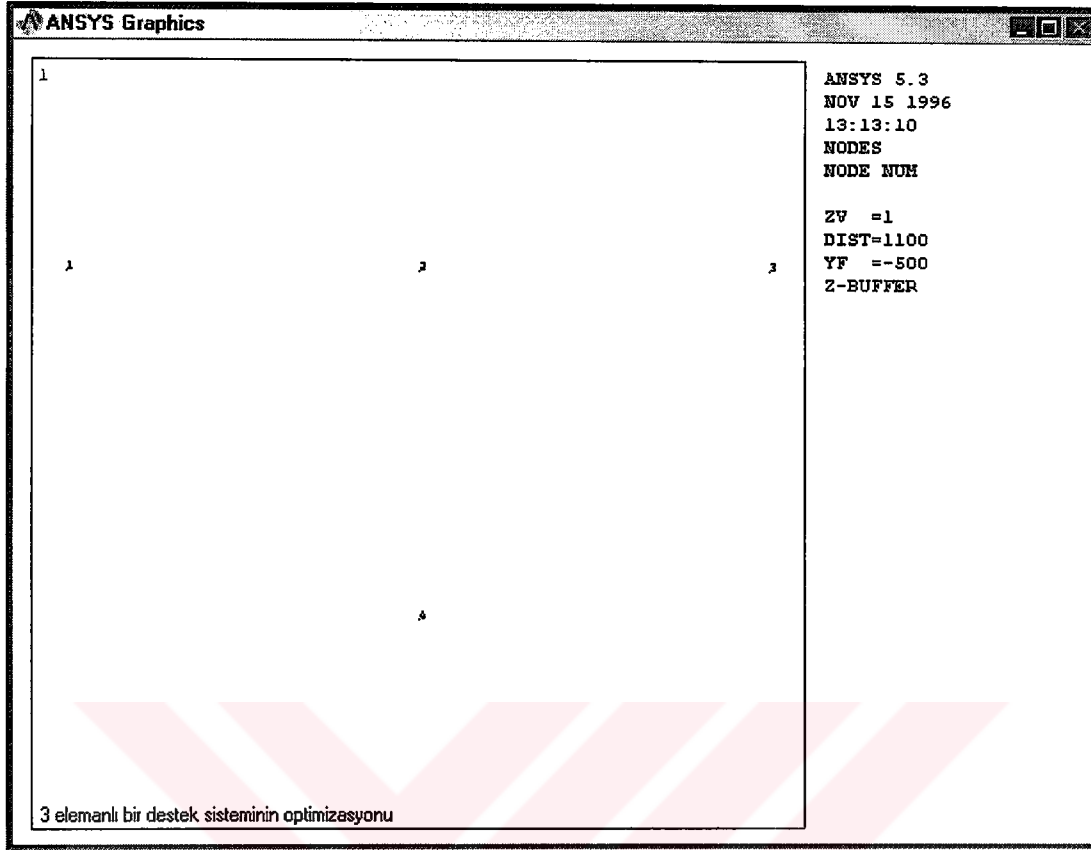
1. “Aktif koordinat sisteminde node oluşturma” diyalog kutusunu (Şekil 4.12) açmak için şu yol izlenir.

Main Menu > Preprocessor > -Modeling-Create- >Nodes >Active CS



Şekil 4.12- Aktif koordinat sisteminde node oluşturma

2. Node numarası için 1 yazılır.
3. X,Y,Z konumları için sırasıyla -B, 0 ve 0 yazılır.
4. “Apply” butonuna basıldığında Node 1 grafik penceresinde görünecektir.
5. Node numarası için 2 yazılır.
6. X,Y,Z konumları için sırasıyla 0, 0 ve 0 yazılır
7. “Apply” butonuna basıldığında Node 2 grafik penceresinde görünecektir
8. Node numarası için 3 yazılır.
9. X,Y,Z konumları için sırasıyla B, 0 ve 0 yazılır.
10. Apply butonuna basıldığında Node 3 Grafik penceresinde görünecektir.
11. Node numarası için 4 yazılır.
12. X,Y,Z konumları için sırasıyla 0, -1000 ve 0 yazılır
13. “OK” butonuna basıldığında diyalog kutusu kapanıp Node 4 Grafik penceresinde görünecektir. (Şekil 4.13)



Şekil 4.13- Çizilen nodların grafik penceresindeki görüntüsü

14. Node numaraları grafik penceresinde görünmüyor ise görünmesi için “numaralandırma ayarları” diyalog kutusunu açmak için şu yol izlenir.

Utility Menu >PlotCtrls >Numbering

15. Node numaraları satırı için “Off” seçilir (yazı değişip “On” olacaktır).

16. Diyalog kutusunu kapatmak için “OK” butonuna basılır.

17. Pencere seçenekleri diyalog kutusunu açmak için şu yol izlenir:

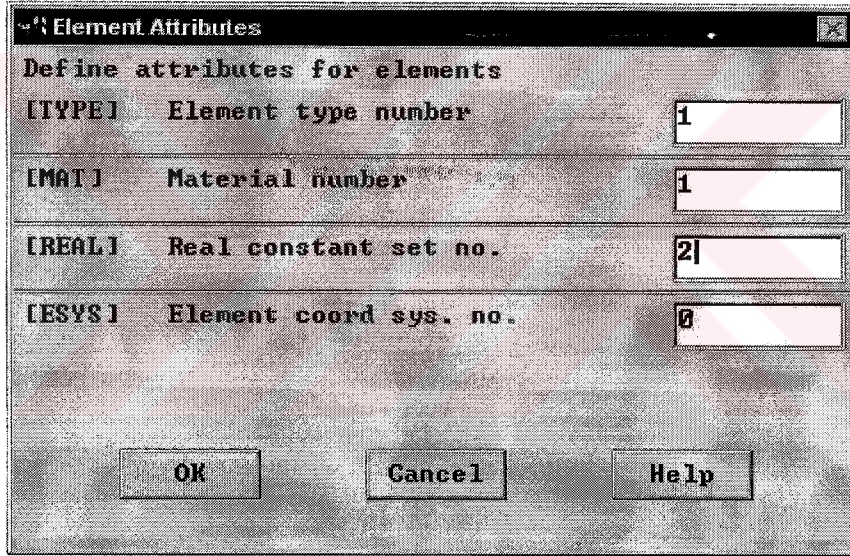
Utility menu >PlotCtrls >Window Controls >Window Options

18. “Location of Triad” seçeneği “Not Shown” olarak değiştirilirse grafik penceresinde X,Y,Z koordinatlarını gösteren şekil kaldırılır. Böylece 0,0,0 noktasındaki 2 numaralı node da görülebilir.

19. “OK” butonuna basılarak diyalog kutusu kapatılabilir.

4.1.8 Adım 8- Elemanların oluşturulması

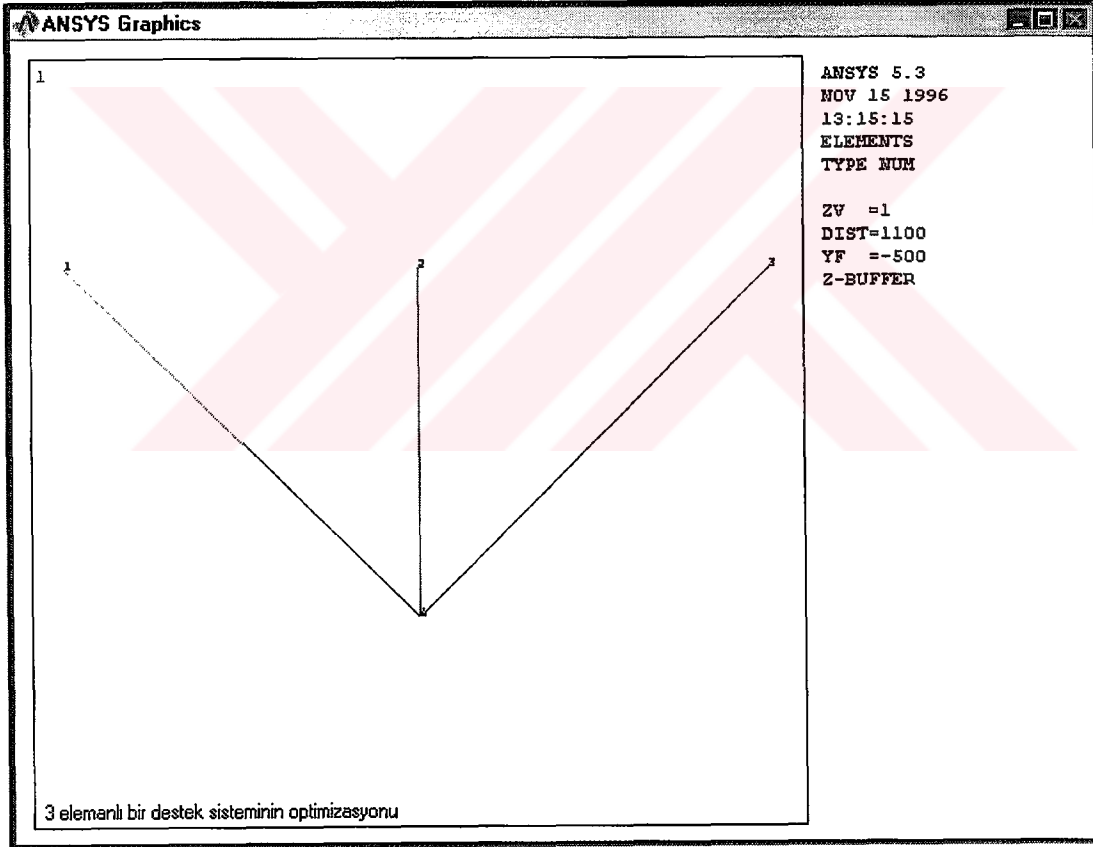
1. “Nodlardan eleman oluşturma” diyalog kutusunu açmak için şu yol izlenir.
Main Menu >Preprocessor >-Modeling-Create >Elements >-Auto Numbered-
Thru Nodes
2. Grafik penceresinde 1 ve 4 numaralı node ‘lar sırayla mouse yardımı ile seçilir.
Seçilen node ‘lar üzerinde küçük kutucuklar görünecektir.
3. “OK” butonuna basılarak diyalog kutusu kapatılır ve Eleman-1 grafik
penceresinde görünür.
4. “Eleman özellikleri” diyalog kutusunu (Şekil 4.14) açmak için şu yol izlenir:
Main Menu >Preprocessor >-Modeling-Create >Elements >Elem Attributes



Şekil 4.14- Elemanların özelliklerinin belirtildiği diyalog kutusu

5. “Gerçek sabit seti numarası” için 2 yazılır.
6. “OK” butonuna basılır.
7. “Nodlardan eleman oluşturma” diyalog kutusunu açmak için şu yol izlenir.
Main Menu >Preprocessor >-Modeling-Create >Elements >-Auto Numbered-
Thru Nodes
8. Grafik penceresinde 2 ve 4 numaralı node ‘lar sırayla mouse yardımı ile seçilir.
9. “OK” butonuna basılarak diyalog kutusu kapatılır ve 2 ve 4 numaralı node ‘lar
arasında oluşan Eleman-2 grafik penceresinde görünür.

10. “Eleman özellikleri” diyalog kutusunu açmak için şu yol izlenir:
Main Menu >Preprocessor >-Modeling>Create >Elements >Elem Attributes
11. “Gerçek sabit seti numarası” için 3 yazılır.
12. “OK” butonuna basılır.
13. “Nodlardan eleman oluşturma” diyalog kutusunu açmak için şu yol izlenir:
Main Menu >Preprocessor >-Modeling>Create >Elements >-Auto Numbered-Thru Nodes
14. Grafik penceresinde 3 ve 4 numaralı node ‘lar sırayla mouse yardımı ile seçilir.
15. “OK” butonuna basılarak diyalog kutusu kapatılır ve 3 ve 4 numaralı node’lar arasında oluşan Eleman-3 grafik penceresinde görünür. (Şekil 4.15)

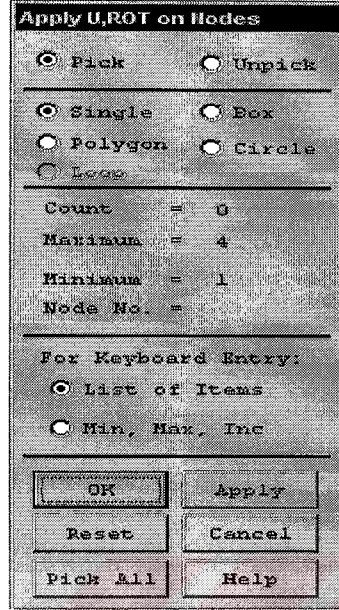


Şekil 4.15- Çizilen elemanların grafik penceresindeki görüntüsü

4.1.9 Adım 9- Yer değiştirme sınırlamaları ve yüklerin uygulanması

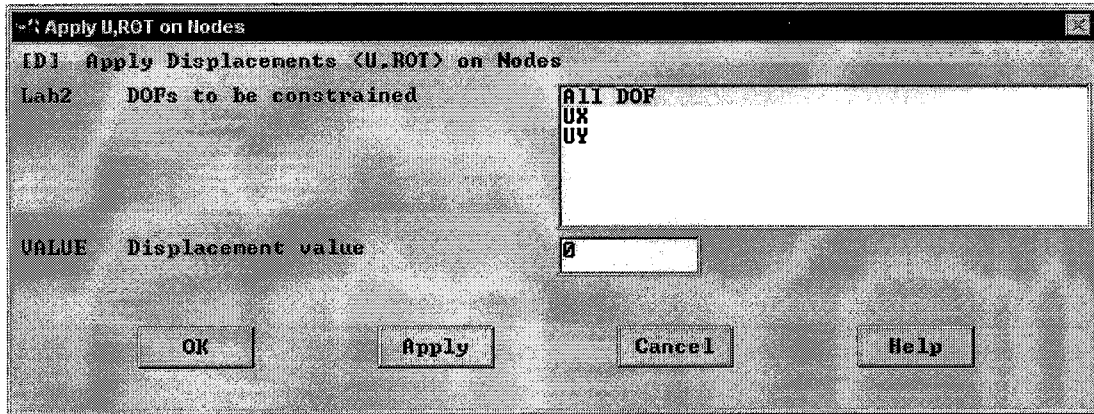
1. “Nodları sınırlama (U,ROT)” diyalog kutusunu (Şekil 4.16) açmak için şu yol izlenir:

Main Menu >Solution >-Loads-Apply >-Structural- >Displacement >On Nodes



Şekil 4.16- Nodları sınırlama diyalog kutusu

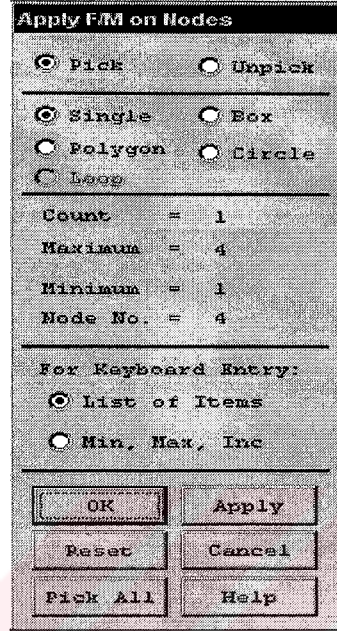
2. ANSYS grafik penceresinde 1,2 ve 3 numaralı nodlar seçilir.
3. “Nodları sınırlama (U,ROT)” diyalog kutusunu kapatmak için “OK” butonuna basılır ve ikinci bir “Nodları sınırlama (U,ROT)” diyalog kutusu açılır.(Şekil 4.17)



Şekil 4.17- Sınırlamaların tipinin belirtildiği diyalog kutusu

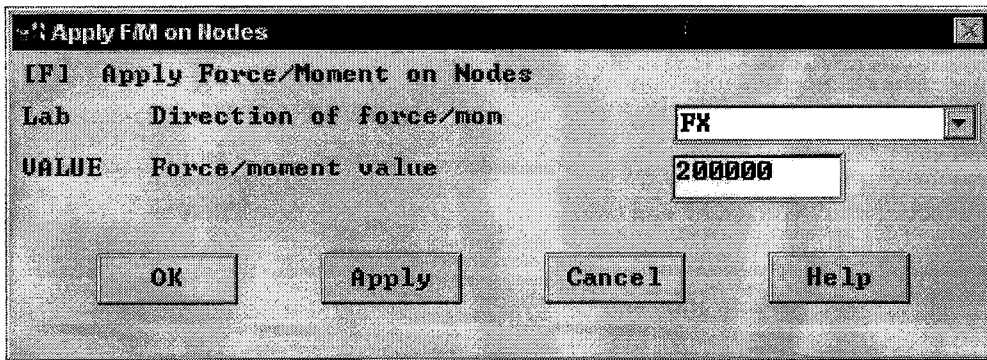
4. Sınırlamalar için “All DOF” seçilir. Böylece seçilen nodlar için her türlü hareket engellenmiş olur.

5. Diyalog kutusu “OK” butonu ile kapatılır.
6. “Nodlara kuvvet uygulama” diyalog kutusunu (Şekil 4.18) açmak için;
Main Menu >Solution >-Loads-Apply >-Structural- >Force/Moment >On Nodes



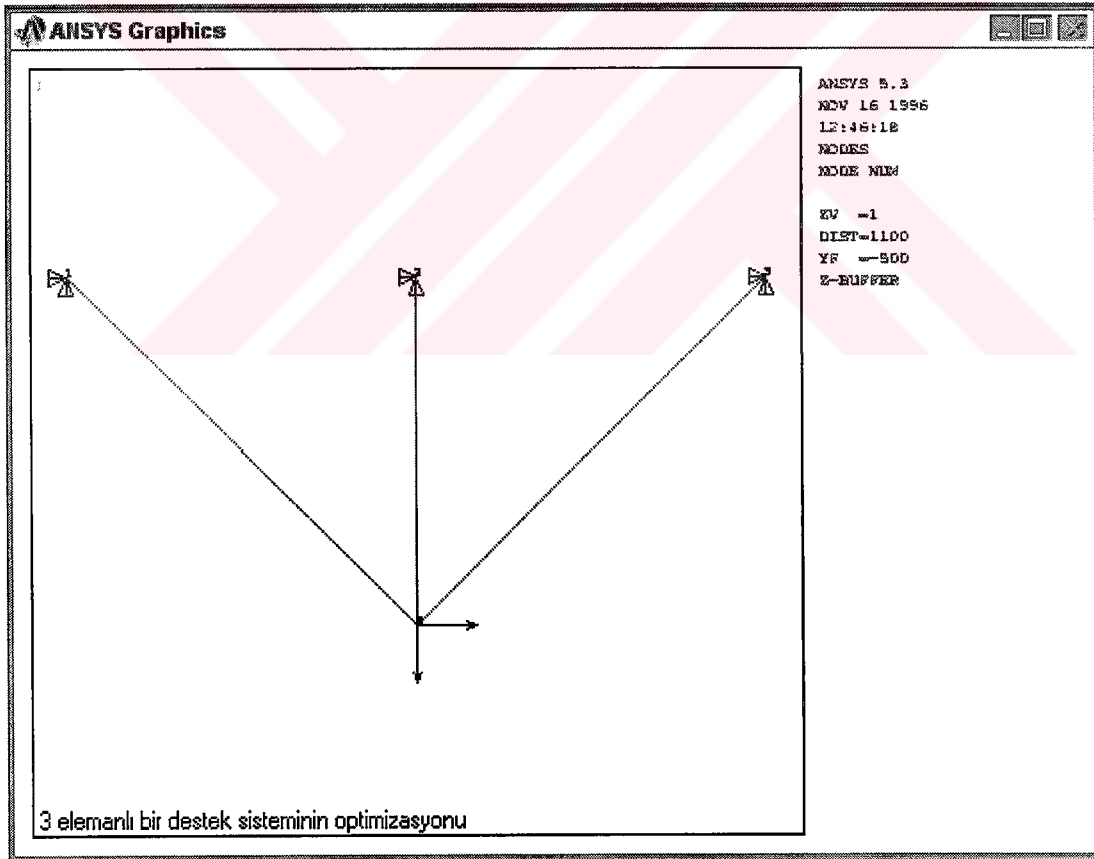
Şekil 4.18- Nodlara kuvvet uygulama diyalog kutusu

7. ANSYS grafik penceresinde Node 4 seçilir.
8. “Nodlara kuvvet uygulama” diyalog kutusunu kapatmak için “OK” butonuna basılır ve ikinci bir “Nodlara kuvvet uygulama” diyalog kutusu açılır. (Şekil 4.19)
9. “Kuvvetin yönü” için FX seçilir.
10. “Kuvvetin büyüklüğü” için 200000 yazılır.



Şekil 4.19- Kuvvetlerin yön ve büyüklüklerinin belirtildiği diyalog kutusu

11. Diyalog kutusu “OK” butonu ile kapatılır. Kuvveti gösteren bir yatay ok Node 4 üzerinde görünür.
12. “Nodlara kuvvet uygulama” diyalog kutusunu açmak için şu yol izlenir:
Main Menu >Solution >-Loads-Apply >-Structural- >Force/Moment >On Nodes
13. ANSYS grafik penceresinde Node 4 seçilir.
14. “Nodlara kuvvet uygulama” diyalog kutusunu kapatmak için “OK” butonuna basılır ve ikinci bir “Nodlara kuvvet uygulama” diyalog kutusu açılır.
15. “Kuvvetin yönü için” için FY seçilir.
16. “Kuvvetin büyüklüğü” için -200000 yazılır.
17. Diyalog kutusu “OK” butonu ile kapatılır. Kuvveti gösteren bir dikey ok Node 4 üzerinde görünür.(Şekil 4.20)



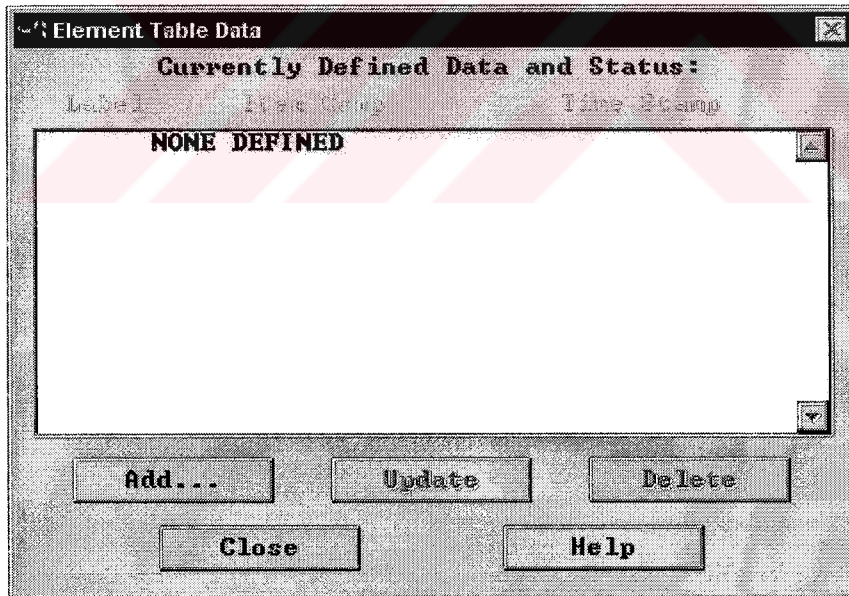
Şekil 4.20- Sınırlama ve kuvvetlerin grafik ekrandaki görüntüleri

4.1.10 Adım 10- Modelin çözülmesi

1. “Aktif yüklerle çözüm” diyalog kutusunu açmak için şu yol izlenir:
Main Menu >Solution >-Solve-Current LS
Çözüm özeti ve yük adımı seçeneklerini özetle gösteren bir durum penceresi açılır.
2. Bilgiler gözden geçirildikten sonra pencere kapatılır.
3. Diyalog kutusunda “OK” butonuna basılır. Böylece çözüm başlar.
4. Çözüm işlemi bittiğinde bir bilgi kutusu ile bu durum belirtilir. Bilgi kutusunu kapatmak için “Close” butonuna basılır.

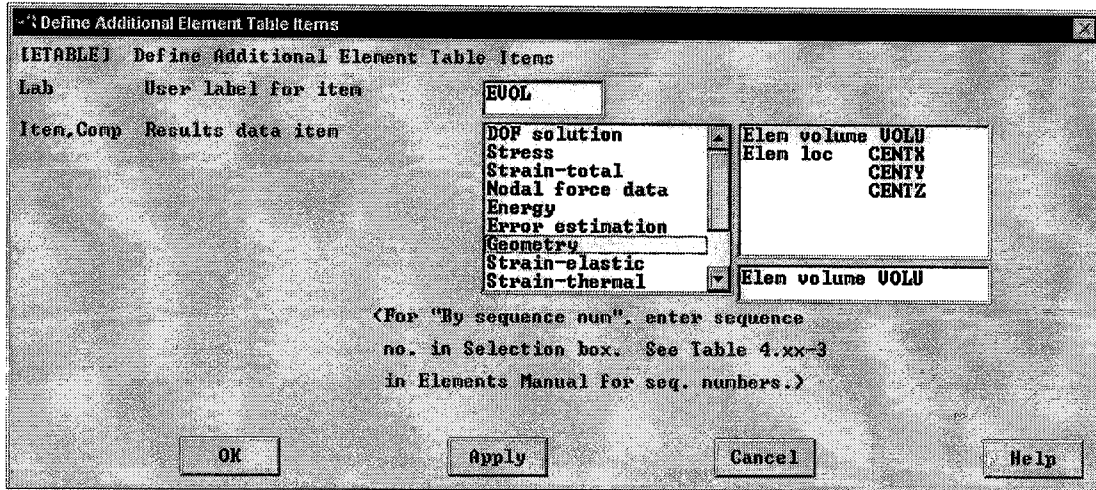
4.1.11 Adım 11- Postprocessor ‘e giriş ve toplam eleman hacminin elde edilmesi

1. “Eleman tablosu bilgisi” diyalog kutusunu (Şekil 4.21) açmak için şu yol izlenir:
Main Menu >General Postproc > Element Table >Define Table



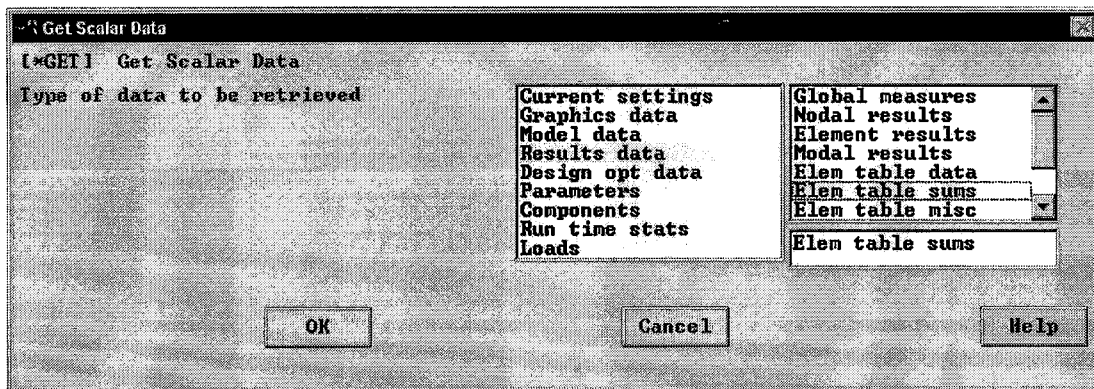
Şekil 4.21- Eleman tablosu bilgisi diyalog kutusu

2. Bir eleman tablosu itemi tanımlamak üzere “Ek eleman tablosu verisi” diyalog kutusunu(Şekil 4.22) açmak için “Add” butonuna basılır.
3. “İtem için kullanıcı etiketi” olarak “EVOL” yazılır. Bu ifade her bir elemanın hacmini belirtmektedir.



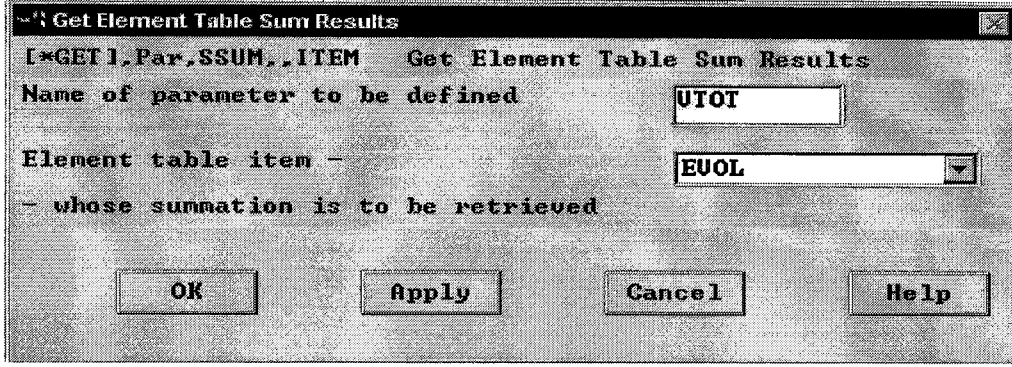
Şekil 4.22- Eleman tablosu oluşturulacak itemin belirtilmesi

4. Sol kolondaki “Geometry” ve sağ kolondaki “VOLU” seçilir.
5. Diyalog kutusunu kapatmak için “OK” butonuna basılır. Tanımlanan büyüklük “Eleman tablosu bilgisi” diyalog kutusunda görünecektir.
6. “Eleman tablosu bilgisi” diyalog kutusunda “Close” butonuna basılır.
7. “Eleman tablosu itemlerinin toplamı” diyalog kutusunu açmak için şu yol izlenir:
Main Menu >General Postproc >Element Table >Sum of Each Item
8. Her üç elemanın hacimlerinin toplamını hesaplatmak için “OK” butonuna basılır. Görünen SSUM komut penceresinde hacim değeri olarak 0.382843E+07 yazacaktır.
9. SSUM komut penceresi kapatılır.
10. “Skalar veri elde etme” diyalog kutusunu (Şekil 4.23) açmak için şu yol izlenir:
Utility Menu >Parameters >Get Scalar Data



Şekil 4.23- Skalar veri elde etme diyalog kutusu

11. Sol kolondaki “Results Data” ve sađ kolondaki “Elem Table Sums” seçilir.
12. Diyalog kutusunu kapatmak ve “Eleman tablosu toplamı sonuçları alma” diyalog kutusunu (Şekil 4.24) açmak için “OK” butonuna basılır.



Şekil 4.24- Eleman tablosu toplamı sonuçlarının elde edilmesi

13. “Tanımlanacak parametrenin ismi” için “VTOT” yazılır.
14. Diyalog kutusunu kapatmak için “OK” butonuna basılır.
15. “Skalar Parametreler” diyalog kutusunu açmak için şu yol izlenir:
Utility Menu >Parameters >Scalar Parameters
16. $RHO=2.85E-4$ (yoğunluk) yazarak “Enter” tuşuna basılır ve parametre listesinde yerini alır.
17. Toplam ağırlığı hesaplatmak üzere $WT=RHO*VTOT$ yazarak “Enter” tuşuna basılır ve parametre listesinde yerini alır.
18. Diyalog kutusunu kapatmak için “Close” butonuna basılır.

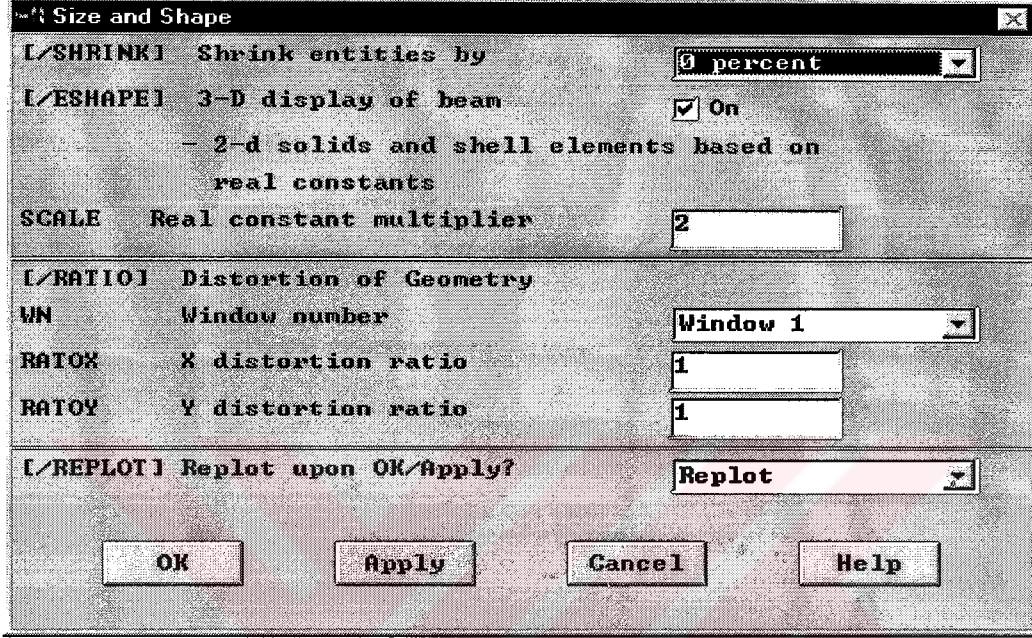
4.1.12 Adım 12- Çubuklarda oluşan eksenel gerilmelerin elde edilmesi

1. “Eleman tablosu verisi” diyalog kutusunu (Şekil 4.21) açmak için şu yol izlenir:
Main Menu >General Postproc >Element Table >Define Table
2. “Ek eleman tablosu verisi” diyalog kutusunu açmak için “Add” butonuna basılır.
3. “İtem için kullanıcı etiketi” için SIG yazılır.
4. “Sonuç verisi itemi” için sol kolondan “By Sequence Num” ve sađ kolondan “LS” seçilir.
5. “Selection” için LS ‘den sonra “1” eklenir, (sıralama numarası olarak).

6. Diyalog kutusunu kapatmak için “OK” butonuna basılır.
7. Eleman tablosu diyalog kutusunda “Close” butonuna basılır.
8. “Skalar veri alma” diyalog kutusunu açmak için şu yol izlenir:
Utility Menu >Parameters >Get Scalar Data
9. Sol kolondaki “Results Data” ve sağ kolondaki “Elem Table Data” seçilir.
10. Diyalog kutusunu kapatmak ve “Eleman tablosu verisi alma” diyalog kutusunu açmak için “OK” butonuna basılır.
11. “Tanımlanacak parametre ismi” için SIG1 (Birinci elemanın gerilmesi) yazılır.
12. “Eleman numarası (N)” için 1 yazılır.
13. “Yeniden denenecek eleman tablosu” için SIG ayarlanır.
14. Diyalog kutusunu kapatıp, “Skalar veri alma” diyalog kutusunu (Şekil 2.23) açmak için “Apply” butonuna basılır.
15. Sol kolondaki “Results Data” ve sağ kolondaki “Elem Table Data” seçilir.
16. “Eleman tablosu verisi alma” diyalog kutusunu açmak için OK butonuna basılır.
17. “Tanımlanacak parametre ismi” için SIG2 (İkinci elemanın gerilmesi) yazılır.
18. “Eleman numarası (N)” için 2 yazılır.
19. “Yeniden denenecek eleman tablosu” için SIG ayarlanır.
20. Diyalog kutusunu kapatıp, “Skalar veri alma” diyalog kutusunu açmak için “Apply” butonuna basılır.
21. Sol kolondaki “Results Data” ve sağ kolondaki “Elem Table Data” seçilir.
22. “Eleman tablosu verisi alma” diyalog kutusunu açmak için “OK” butonuna basılır.
23. “Tanımlanacak parametre ismi” için SIG3 (Üçüncü elemanın gerilmesi) yazılır.
24. “Eleman numarası (N)” için 3 yazılır.
25. “Yeniden denenecek eleman tablosu” için SIG ayarlanır.
26. Diyalog kutularını kapatmak için “OK” butonuna basılır.
27. Şimdi eksenel gerilmelerin mutlak değerlerini hesaplamak üzere “Skalar Parametreler” diyalog kutusunu açmak için şu yol izlenir:
Utility Menu >Parameters >Scalar Parameters
28. SIG1=ABS(SIG1) yazılıp Enter’e basılır.
29. SIG2=ABS(SIG2) yazılıp Enter’e basılır.
30. SIG3=ABS(SIG3) yazılıp Enter’e basılır.
31. “Skalar Parametreler” diyalog kutusunu kapatmak için “Close” butonuna basılır.

4.1.13 Adım 13- Halihazırdaki dizaynın görüntülenmesi

1. “Büyüklik ve şekil” diyalog kutusunu (Şekil 4.25) açmak için şu yol izlenir:
Utility Menu >PlotCtrls>Style >Size&Shape



Şekil 4.25- Şekil ve büyüklüğün düzenlenebildiği diyalog kutusu

2. “Kirişin üç boyutlu görüntüsü” seçilir (OFF değişerek ON olacaktır).
3. “Gerçek sabit çarpanı” için 2 yazılır.
4. Diyalog kutusunu kapatmak için “OK” butonuna basılır.
5. “Pan, Zoom, Rotate” diyalog kutusunu açmak için şu yol izlenir:
Utility Menu >PlotCtrls>Pan,Zoom,Rotate
6. 3-Çubuklu destek sisteminin izometrik görünüşünü elde etmek için ISO butonuna basılır.
7. Diyalog kutusunu kapatmak için “Close” butonuna basılır.
8. Destek sistemini çizdirmek için şu yol izlenir:
Utility Menu >Plot >Elements

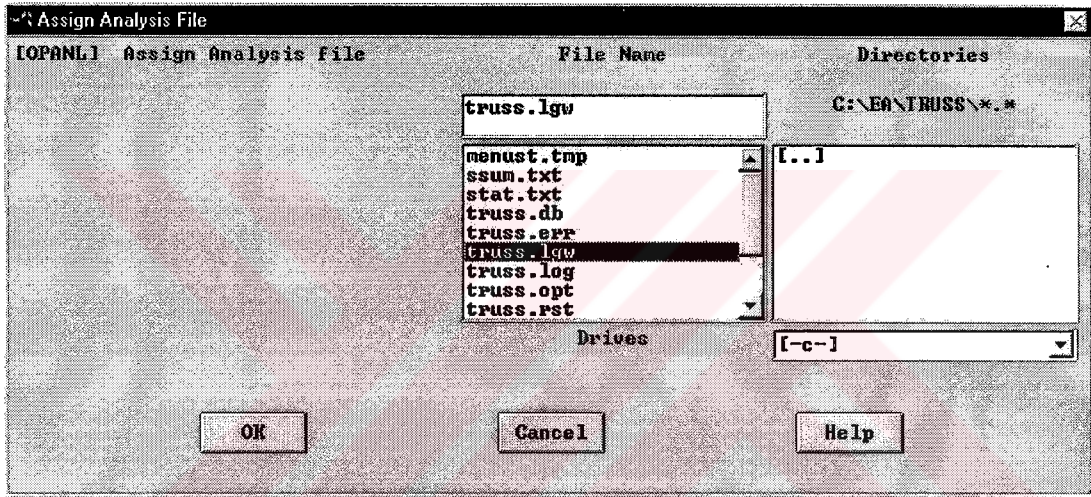
4.1.14 Adım 14- Optimizasyon için analiz dosyasının oluşturulması

1. “Veritabanı Log dosyası yazma” diyalog kutusunu açmak için şu yol izlenir:
Utility Menu >File >Write DB Log File

2. Dosya adı için “truss.lgw” yazılır.
3. Diyalog kutusunu kapatmak için “OK” butonuna basılır.

4.1.15 Adım 15- Optimizer’a giriş ve analiz dosyasının tanıtılması

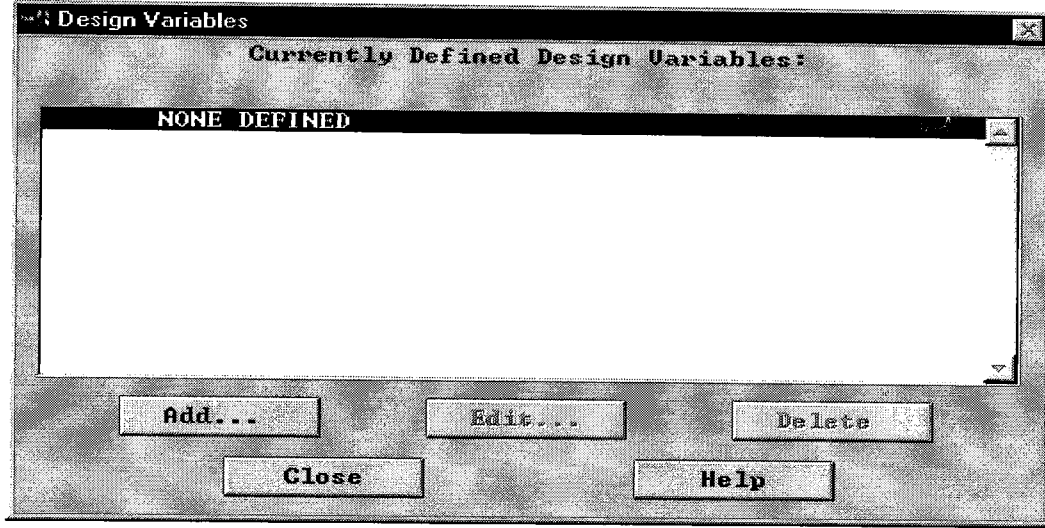
1. “Analiz dosyası atama” diyalog kutusunu (Şekil 4.26) açmak için şu yol izlenir:
Main Menu >Design Opt > -Analysis File-Assign
2. Listedeki “truss.lgw” dosyası seçilir.
3. Diyalog kutusunu kapatmak için “OK” butonuna basılır.



Şekil 4.26- Analiz dosyasının belirtilmesi

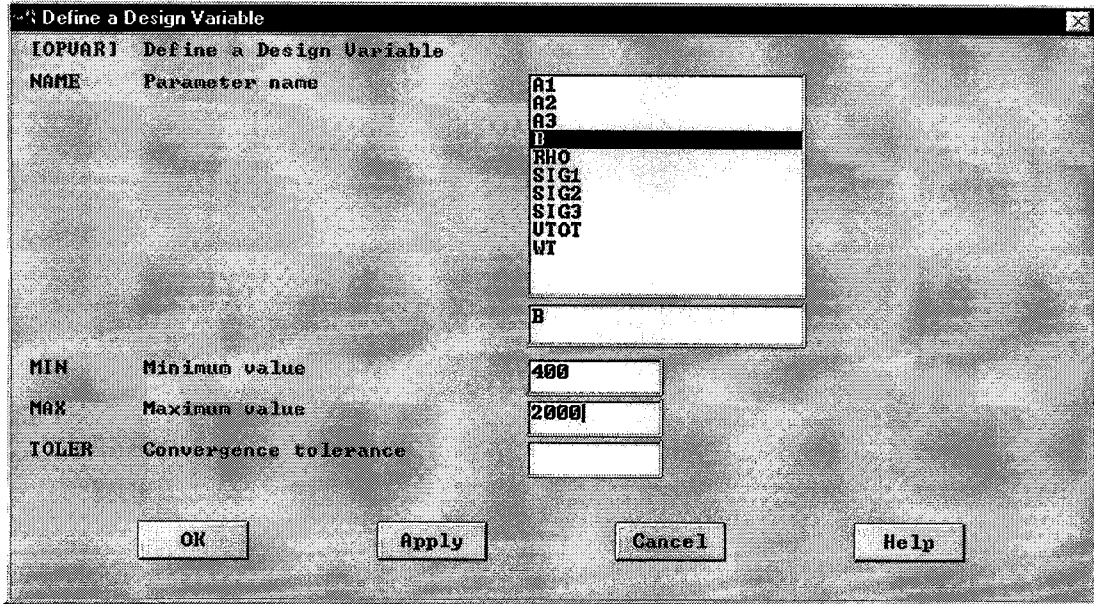
2.5.16 Adım 16- Optimizasyon Dizayn Değişkenlerinin Tanımlanması

1. “Dizayn değişkenleri” diyalog kutusunu (Şekil 4.27) açmak için şu yol izlenir:
Main Menu >Design Opt >Design Variables
2. “Bir dizayn değişkeni tanımla” diyalog kutusunu (Şekil 4.28) açmak için “Add” butonuna basılır.
3. “Parametre ismi” listesinden B seçilir.
4. “Minimum değer” için 400 yazılır.
5. “Maksimum değer” için 2000 yazılır.
6. “Apply” butonuna basılarak B için tanımlama işlemi tamamlanır.
7. “Parametre ismi” listesinden A1 seçilir.



Şekil 4.27- Dizayn değişkenleri diyalog kutusu

8. "Minimum değer" için 1 yazılır.
9. "Maksimum değer" için 1000 yazılır.
10. "Apply" butonuna basılarak A1 için tanımlama işlemi tamamlanır.
11. "Parametre ismi" listesinden A2 seçilir.
12. "Minimum değer" için 1 yazılır.
13. "Maksimum değer" için 1000 yazılır.
14. "Apply" butonuna basılarak A2 için tanımlama işlemi tamamlanır.

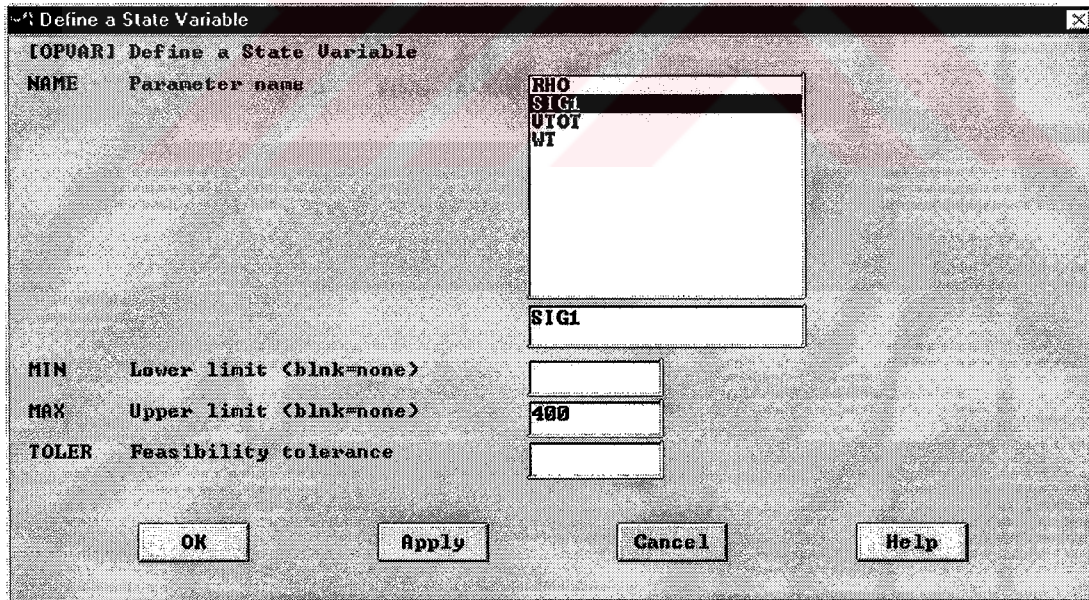


Şekil 4.28- Dizayn değişkenlerinin tanımlandığı diyalog kutusu

15. "Parametre ismi" listesinden A3 seçilir.
16. "Minimum değer" için 1 yazılır.
17. "Maksimum değer" için 1000 yazılır.
18. Diyalog kutusunu kapatmak için "OK" butonuna basılır. Tanımlanan bütün dizayn değişkenleri Şekil 4.27 de ki tabloda yerini alacaktır.
19. "Dizayn değişkenleri" diyalog kutusunu kapatmak için "Close" butonuna basılır.

4.1.17 Adım 17- Sınırlamaların (SVs) tanımlanması

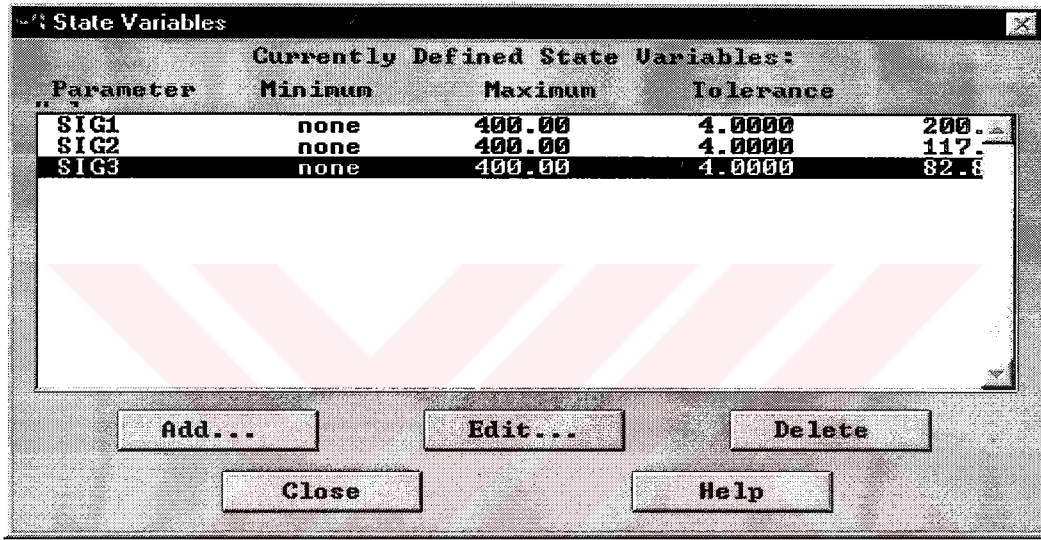
1. SVs diyalog kutusunu açmak için şu yol izlenir:
Main Menu >Design Opt >State Variables
2. "Değişken tanımla" diyalog kutusunu (Şekil 4.29) açmak için "Add" butonuna basılır.
3. Parametre listesinden SIG1 seçilir.
4. Üst Limit için 400 yazılır.



Şekil 4.29- SVs tanımlanması

5. "Apply" butonuna basılarak SIG1 için tanımlama işlemi tamamlanır.
6. Parametre listesinden SIG2 seçilir.
7. Üst Limit için 400 yazılır.

8. “Apply” butonuna basilarak SIG2 için tanımlama işlemi tamamlanır.
9. Parametre listesinden SIG3 seçilir.
10. Üst Limit için 400 yazılır.
11. Diyalog kutusunu kapatmak için “OK” butonuna basılır.
12. Tanımlanan bütün sınırlamalar state variables diyalog kutusunda yerini alacaktır(Şekil 4.30). “SVs” diyalog kutusunu kapatmak için “Close” butonuna basılır.



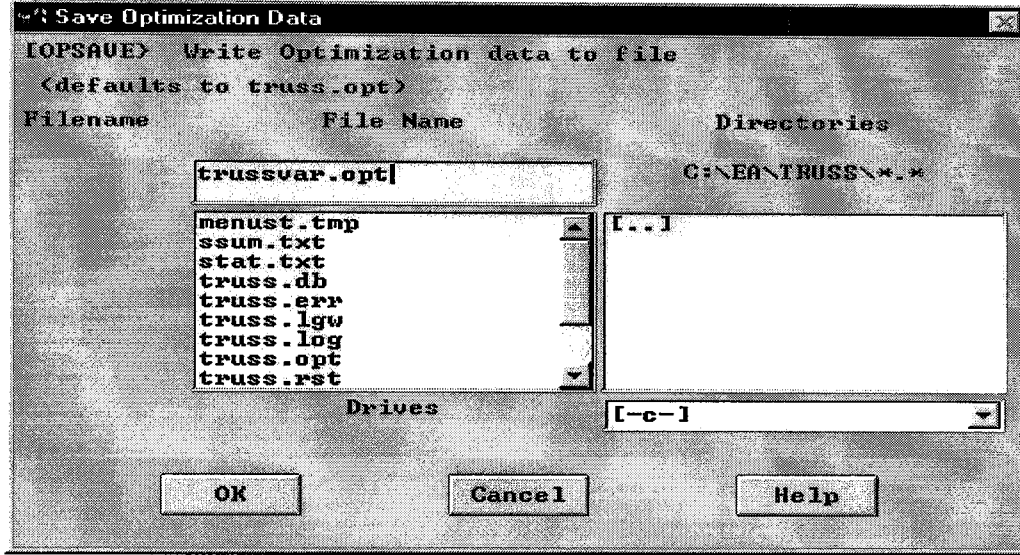
Şekil 4.30- Tanımlanan SVs

4.1.18 Adım 18- Optimizasyon veritabanının saklanması

1. “Optimizasyon veritabanını sakla” diyalog kutusunu (Şekil 4.31) açmak için şu yol izlenir:

Main Menu >Design Opt >-Opt Database-Save

2. “trussvar.opt” yazılır.
3. Diyalog kutusunu kapatmak için “OK” butonuna basılır.



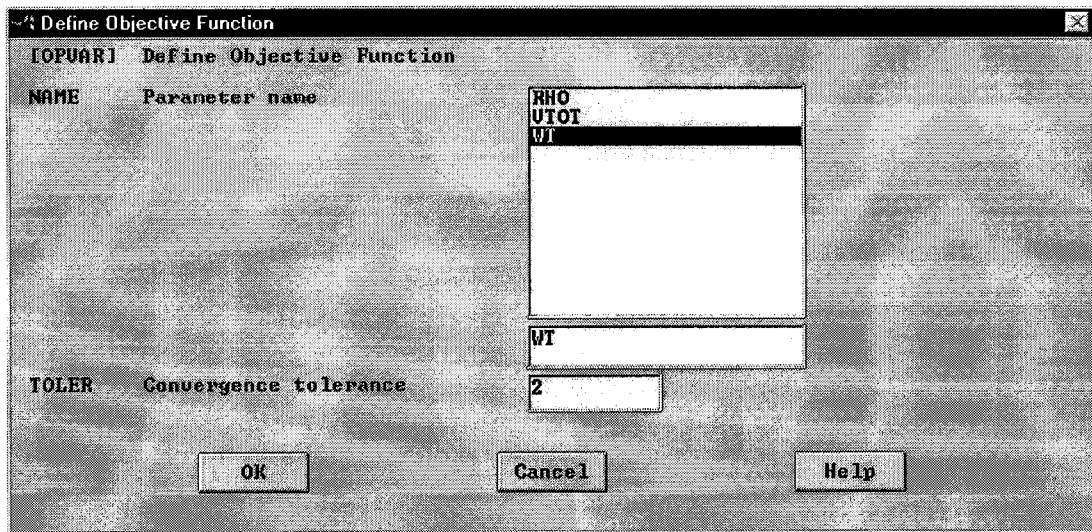
Şekil 4.31- Oluşturulan optimizasyon bilgilerinin saklanması

4.1.19 Adım 19- Ağırlığın hedef fonksiyonu olarak tanımlanması

1. “Hedef fonksiyonunu tanımla” diyalog kutusunu (Şekil 4.32) açmak için şu yol izlenir:

Main Menu >Design Opt >Objective

2. Parametre listesinden WT seçilir.
3. “Yaklaşım toleransı” için 2 yazılır.
4. Diyalog kutusunu kapatmak için “OK” butonuna basılır.



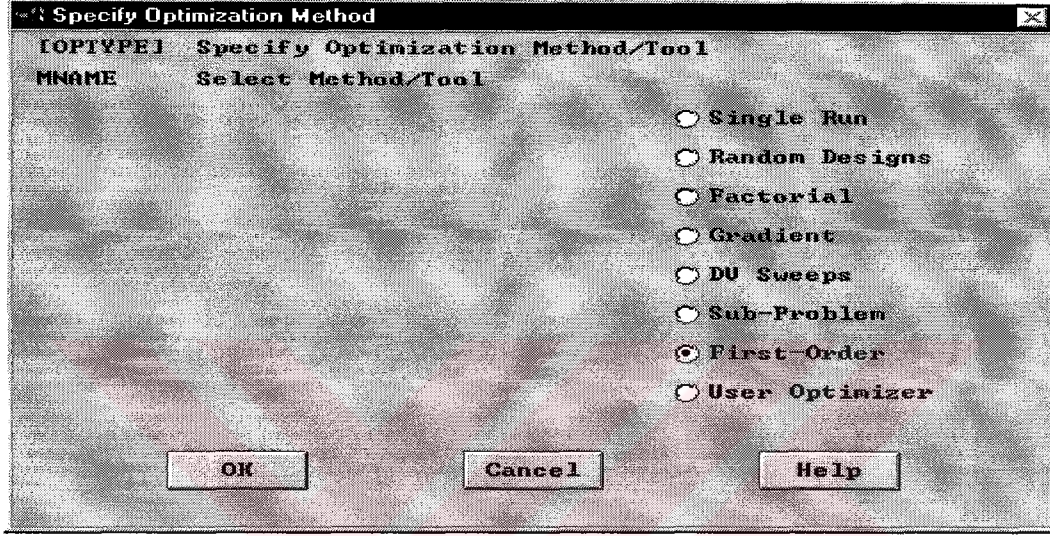
Şekil 4.32- Hedef fonksiyonunun tanımlanması

4.1.20 Adım 20- Birinci merteye optimizasyon metodunun belirtilmesi

1. “Optimizasyon metodunu belirtme” diyalog kutusunu (Şekil 4.33) açmak için şu yol izlenir:

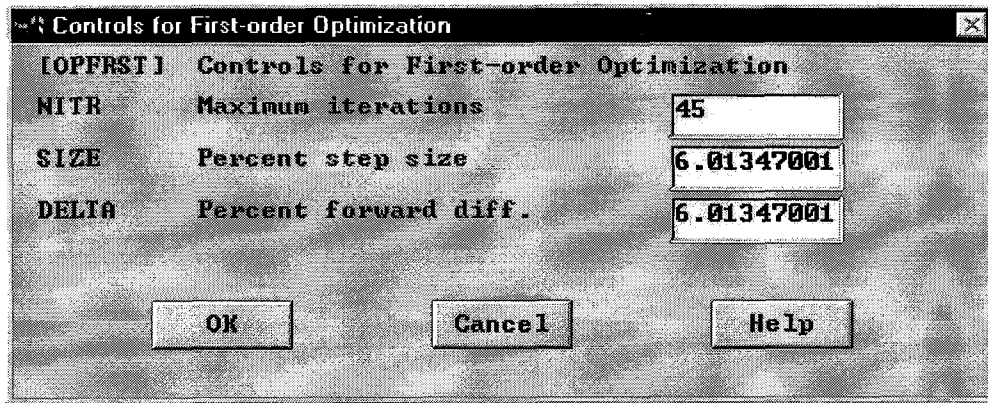
Main Menu >Design Opt >Method/Tool

2. Method/Tool listesinden “First-Order” seçilir.



Şekil 4.33- Optimizasyon metodunun tanımlanması

3. “Birinci merteye optimization metodu için ayarlar” diyalog kutusunu (Şekil 4.34) açmak için “OK” butonuna basılır.
4. “Maksimum iterasyon sayısı” için 45 yazılır.
5. Diyalog kutusunu kapatmak için “OK” butonuna basılır.



Şekil 4.34- Birinci merteye metodu kontrolleri

4.1.21 Adım 21- Optimizasyonun çalıştırılması

1. “Optimizasyonun çalıştırılması” diyalog kutusunu açmak için şu yol izlenir:
Main Menu >Design Opt >Run
2. Analiz ile ilgili bilgileri gözden geçirdikten sonra ”OK” butonuna basılarak işlem başlatılır.

Çözüm işlemi biraz zaman alacaktır. Optimizasyon döngülerinin sonucunda, en iyi dizayn seti 16 numaralı dizayn setidir ve optimum ağırlık 301.23 olacaktır.

4.1.22 Adım 22- En İyi Dizayn Seti ve Tüm Dizayn Setlerinin Listelenmesi

1. “Dizayn setlerini listeleme” diyalog kutusunu açmak için şu yol izlenir:
Main Menu >Design Opt >-Design Sets-List
2. BEST Set seçilir.
3. OPLIST komut penceresini açmak için “OK” butonuna basılır. En iyi set 16 dır. Yani 16. iterasyonda elde edilen parametre değerleri optimum dizaynı vermektedir. Bu set için değişken değerleri Tablo 4.1’de verilmiştir.

Tablo 4.1- En iyi dizayn seti

SET 16		
(FEASIBLE)		
SIG1	(SV)	403.94
SIG2	(SV)	383.24
SIG3	(SV)	65.836
B	(DV)	1125.6
A1	(DV)	662.13
A2	(DV)	58.473
A3	(DV)	1.0000
WT	(OBJ)	301.23

4. Optimizasyon parametrelerinin değerlerini inceledikten sonra pencere kapatılabilir.
5. Oluşan bütün setleri incelemek için şu yol izlenir:
Main Menu >Design Opt >-Design Sets-List

Tablo 4.2- Uygulama-1 için oluşan bütün dizayn setleri

	SET 1	SET 2	SET 3	SET 4	SET 5	SET 6
	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)
SIG1 (SV)	200.00	365.04	385.14	389.52	367.70	383.57
SIG2 (SV)	117.16	98.942	119.88	135.92	248.69	301.40
SIG3 (SV)	82.843	205.16	213.19	213.75	159.07	141.67
B (DV)	1000.0	487.54	627.97	739.25	1176.4	1221.5
A1 (DV)	1000.0	800.38	682.75	645.58	677.72	673.50
A2 (DV)	1000.0	858.85	609.88	486.43	189.85	120.91
A3 (DV)	1000.0	800.38	530.61	397.55	83.558	1.0000
WT (OBJ)	1091.1	752.32	582.15	508.34	389.11	337.92
	SET 7	SET 8	SET 9	SET 10	SET 11	SET 12
	(FEASIBLE)	(FEASIBLE)	(INFEASIBLE)	(INFEASIBLE)	(FEASIBLE)	(FEASIBLE)
SIG1 (SV)	396.54	393.06	> 409.68	> 408.97	401.00	399.99
SIG2 (SV)	328.46	379.01	361.34	379.09	358.81	381.45
SIG3 (SV)	128.21	82.610	105.26	89.092	96.892	74.794
B (DV)	1203.4	1200.7	1172.2	1170.6	1166.1	1160.1
A1 (DV)	655.45	661.98	641.45	642.97	656.79	659.94
A2 (DV)	103.42	88.475	81.680	77.184	79.734	72.634
A3 (DV)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
WT (OBJ)	322.20	320.46	305.39	304.55	310.70	309.21
	SET 13	SET 14	SET 15	SET 16	SET 17	SET 18
	(INFEASIBLE)	(INFEASIBLE)	(INFEASIBLE)	(FEASIBLE)	(INFEASIBLE)	(INFEASIBLE)
SIG1 (SV)	> 414.23	> 413.76	> 404.51	403.94	> 413.55	> 413.49
SIG2 (SV)	362.44	380.49	362.77	383.24	338.44	386.32
SIG3 (SV)	97.581	80.988	86.168	65.836	108.81	64.864
B (DV)	1135.4	1134.4	1131.0	1125.6	1105.1	1102.8
A1 (DV)	643.14	644.18	659.77	662.13	651.98	652.77
A2 (DV)	66.175	62.543	64.169	58.473	56.613	48.500
A3 (DV)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
WT (OBJ)	296.62	295.88	302.59	301.23	293.49	291.20
	SET 19	SET 20	SET 21			
	(INFEASIBLE)	(INFEASIBLE)	(INFEASIBLE)			
SIG1 (SV)	> 421.33	> 412.45	> 409.61			
SIG2 (SV)	229.65	311.09	363.13			
SIG3 (SV)	197.61	104.19	50.519			
B (DV)	1026.2	1009.1	1011.2			
A1 (DV)	662.33	682.41	686.58			
A2 (DV)	23.426	6.2941	6.2941			
A3 (DV)	1.0000	1.0000	1.0000			
WT (OBJ)	277.56	278.50	280.48			

6. ALL Sets seçilir.
7. OPLIST komut penceresini açmak için “OK” butonuna basılır. Böylece elde edilen bütün setler listelenmiş olur. (Tablo 4.2)
8. Dizayn setleri inceledikten sonra pencere kapatılabilir.

4.1.23 Adım 23- Ağırlığın (WT) ve taban uzunluğunun (B) iterasyon sayısının bir fonksiyonu olarak görüntülenmesi

1. “Pan-Zoom-Rotate” diyalog kutusunu açmak için şu yol izlenir:
Utility Menu >PlotCtrls >Pan,Zoom,Rotate
2. X-Y düzleminde bir görünüş için “Front” butonuna basılır.
3. Diyalog kutusunu kapatmak için “Close” butonuna basılır.
4. “Grafik kontrolleri” diyalog kutusunu (Şekil 4.35) açmak için şu yol izlenir:
Utility Menu >PlotCtrls >Style >Graphs
5. “x eksen etiketi” için ITERASYON SAYISI yazılır. (Türkçe karakter kullanılmaz).
6. “y eksen etiketi” için SISTEMIN AGIRLIGI yazılır.
7. Diyalog kutusunu kapatmak için “OK” butonuna basılır.
8. “Dizayn seti parametrelerinin grafiğinin çizimi” diyalog kutusunu (Şekil 4.36) açmak için şu yol izlenir:
Main Menu >Design Opt >Graphs/Tables
9. “y eksen parametresi” için WT seçilir.
10. Diyalog kutusunu kapatmak için “OK” butonuna basılır. ANSYS Grafik penceresinde ağırlık ile iterasyon sayısı arasındaki grafik çizilecektir. (Şekil 4.37)
11. “Grafik kontrolleri” diyalog kutusunu açmak için şu yol izlenir:
Utility Menu >PlotCtrls >Style >Graphs
12. “y eksen etiketi” için TABAN UZUNLUGU yazılır.
13. Diyalog kutusunu kapatmak için “OK” butonuna basılır.

Graph Controls

[/GROPT],VIEW Viewing control ☐ Off

[/GTHK] Thickness of curves

[/GROPT],FILL Area under curve

[/GRID] Type of grid

[/GTHK] Thickness of grid

[/GROPT],CGRID Display grid - ☐ Off
- superimposed on filled curves

[/AXLAB] X-axis label

[/AXLAB] Y-axis label

[/GTHK] Thickness of axes

[/GRITYP] Number of Y-axes

[/XRANGE] X-axis range

☒ Auto calculated
☐ Specified range

XMIN,XMAX Specified X range

[/YRANGE] Y-axis range

☒ Auto calculated
☐ Specified range

YMIN,YMAX Specified Y range -

NUM - for Y-axis number

[/GROPT],ASCAL Y ranges for -

OK Apply Cancel Help

Şekil 4.35- Grafik kontrolleri diyalog kutusu

Graph/List Tables of Design Set Parameters

Graph (or List) Y-Variable Parameters vs. X-Variable Parameters

[XVAROPT] X-variable parameter

Set number
A1
A2
A3
B
RHO
SIG1
SIG2
SIG3
UTOT
WT

Set number

<Design sets will be sorted by X-variable parameter>

[PLUAROPI/PRUAROPI] Y-Variable Parameters to be Graphed/Listed

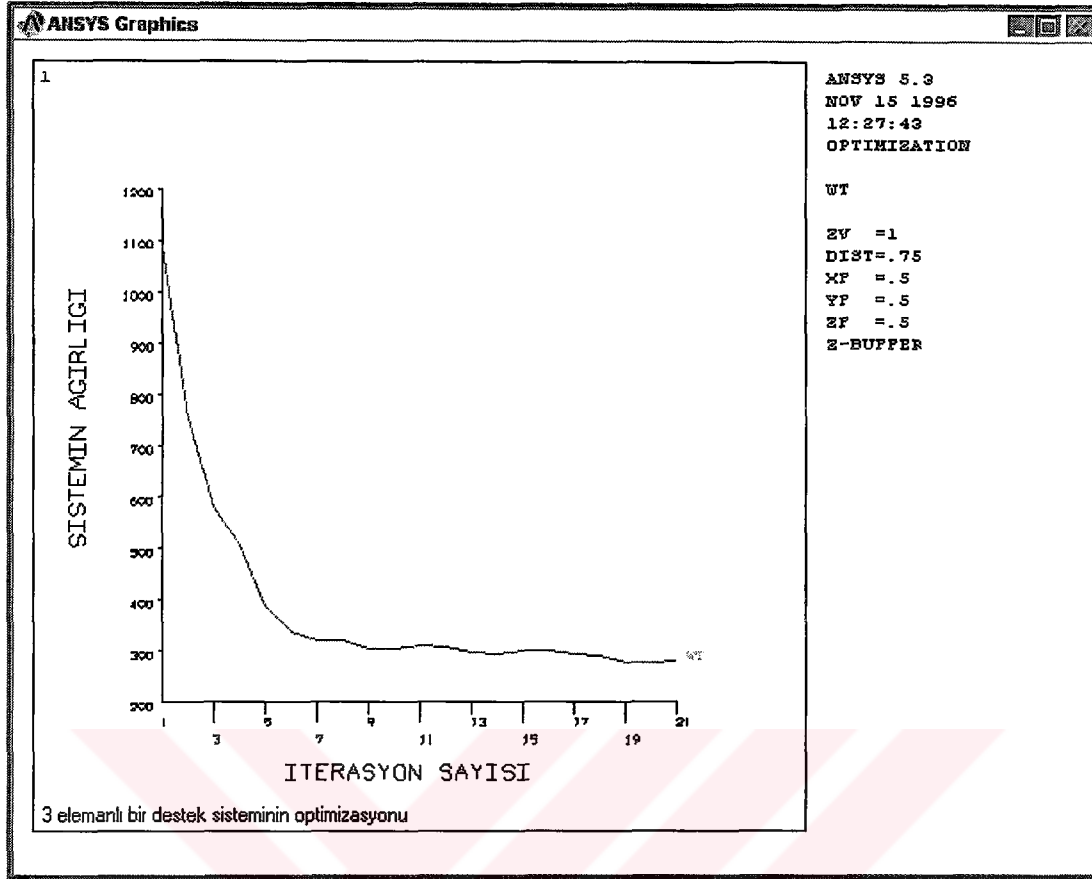
NUAR Y-variable params << 11>

B
RHO
SIG1
SIG2
SIG3
UTOT
WT

Graph or List Table?

OK Apply Cancel Help

Şekil 4.36- Grafikte yer alacak parametrelerin belirtilmesi



Şekil 4.37- İterasyon sayısına bağlı olarak sistemin ağırlığında meydana gelen değişim

14. “Dizayn seti parametrelerinin grafiğinin çizimi” diyalog kutusunu açmak için şu yol izlenir:

Main Menu >Design Opt >Graphs/Tables

15. “y eksen parametresi” için seçili olan WT iptal edilip B seçilir.

16. Diyalog kutusunu kapatmak için “OK” butonuna basılır. ANSYS Grafik penceresinde taban uzunluğu ile iterasyon sayısı arasındaki grafik çizilecektir.

(Şekil 4.38)

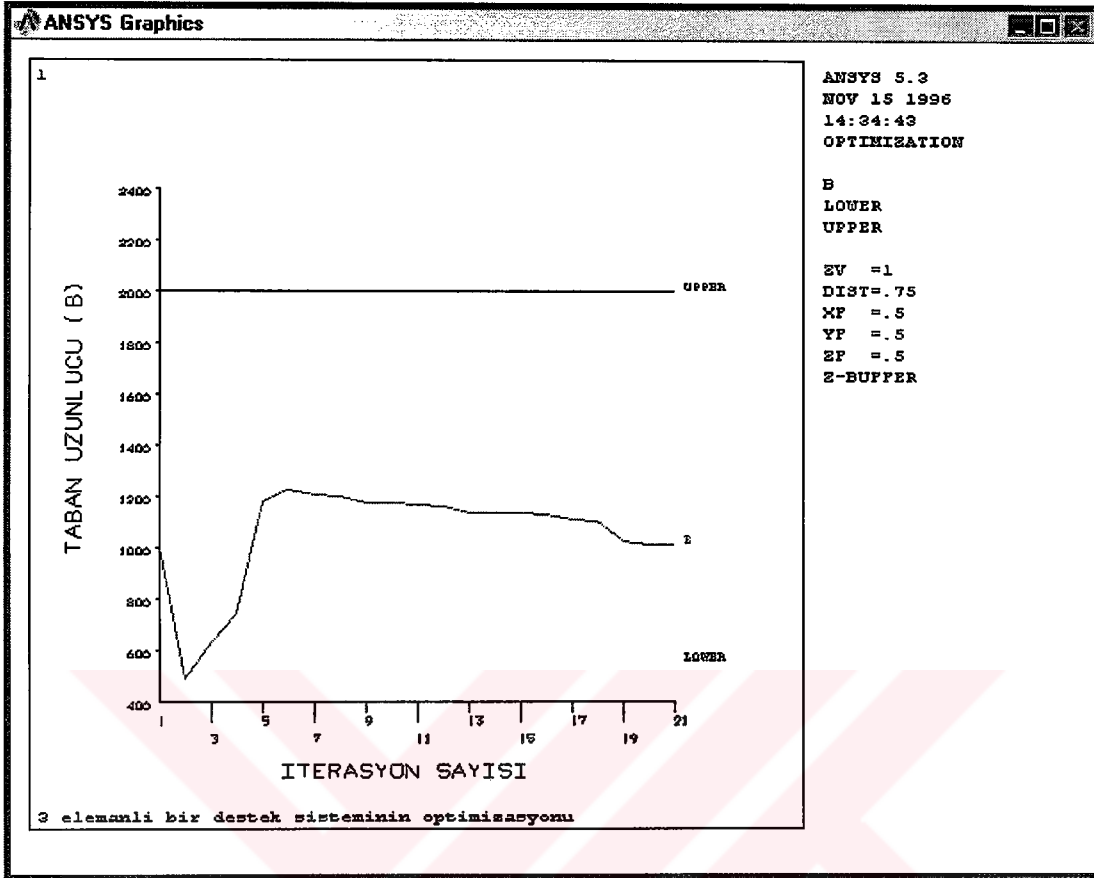
4.1.24 Adım 24- Maksimum gerilme ve kesit alanının iterasyon sayısının bir fonksiyonu olarak görüntülenmesi

1. “Grafik kontrolleri” diyalog kutusunu açmak için şu yol izlenir:

Utility Menu >PlotCtrls >Style >Graphs

2. “y eksen etiketi” için MAKSIMUM GERİLME yazılır.

3. Diyalog kutusunu kapatmak için “OK” butonuna basılır.

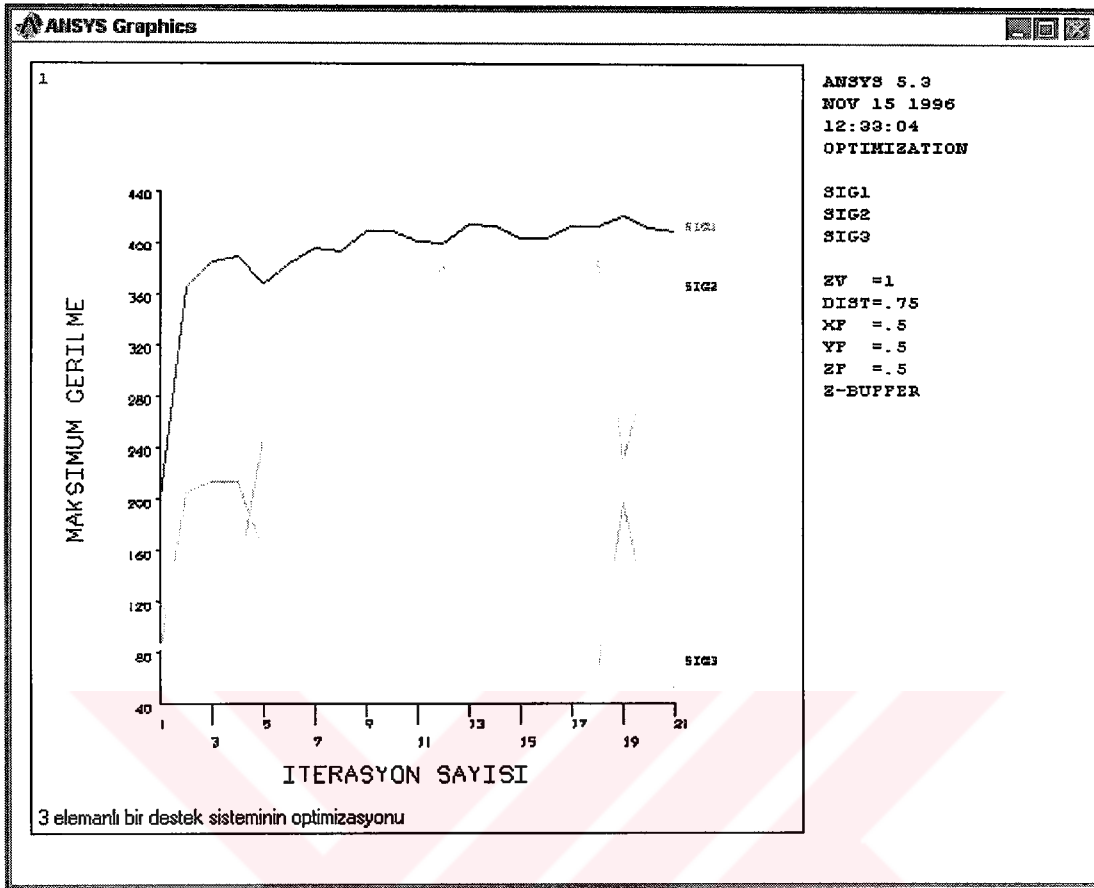


Şekil 4.38- İterasyon sayısına bağlı olarak taban uzunluğunda meydana gelen değişim

4. “Dizayn seti parametrelerinin grafiğinin çizimi” diyalog kutusunu açmak için şu yol izlenir:

Main Menu >Design Opt >Graphs/Tables

5. “y eksen parametreleri” için daha önceden seçilmiş olan B iptal edilip SIG1, SIG2 ve SIG3 seçilir.
6. Diyalog kutusunu kapatmak için “OK” butonuna basılır. ANSYS Grafik penceresinde çubuklardaki gerilme ile iterasyon sayısı arasındaki grafik çizilecektir. (Şekil 4.39)
7. “Grafik kontrolleri” diyalog kutusunu açmak için şu yol izlenir:
Utility Menu >PlotCtrls >Style >Graphs
8. “y eksen etiketi” için KESIT ALANI yazılır.
9. Diyalog kutusunu kapatmak için “OK” butonuna basılır.



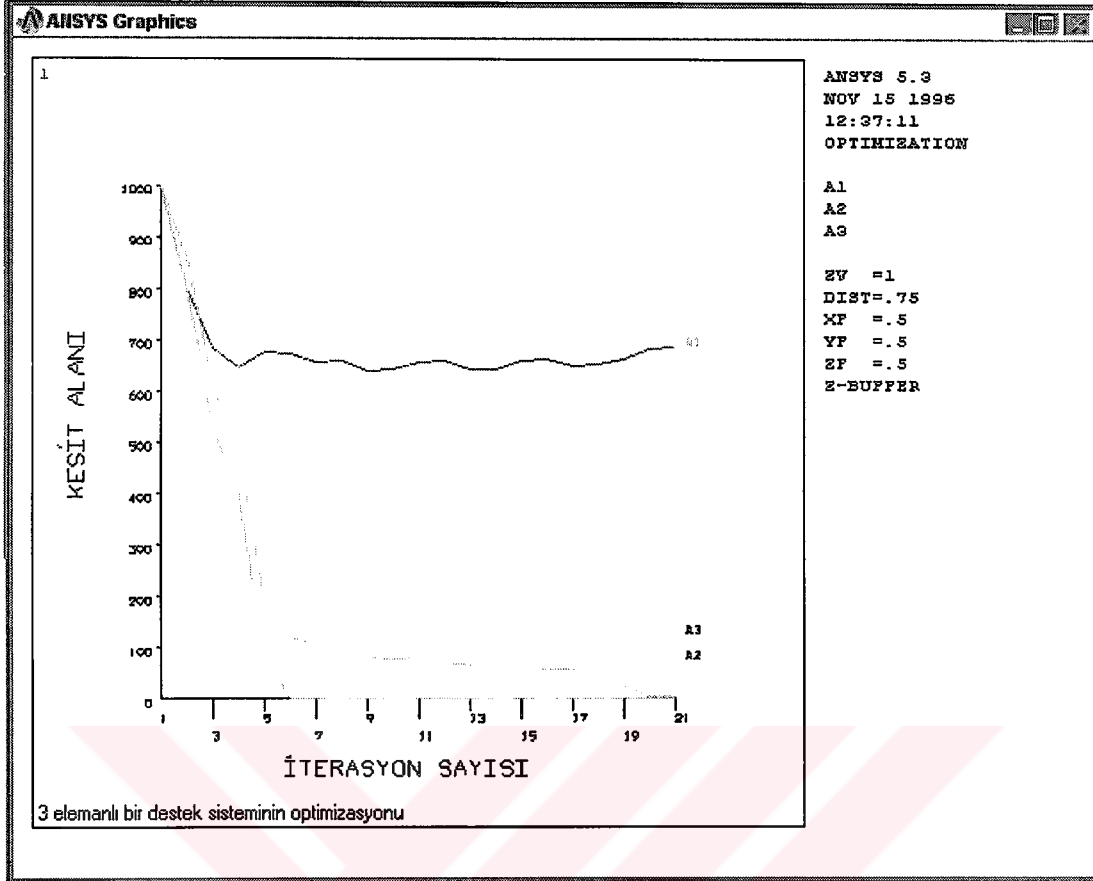
Şekil 4.39- İterasyon sayısına bağlı olarak elemanlarda oluşan maksimum gerilmede meydana gelen değişim

10. “Dizayn seti parametrelerinin grafiğinin çizimi” diyalog kutusunu açmak için şu yol izlenir:

Main Menu >Design Opt >Graphs/Tables

11. “y eksen parametreleri” için seçilmiş plan SIG1, SIG2 ve SIG3 iptal edilip A1, A2 ve A3 seçilir.

12. Diyalog kutusunu kapatmak için “OK” butonuna basılır. ANSYS Grafik penceresinde çubukları kesit alanı ile iterasyon sayısı arasındaki grafik çizilecektir. (Şekil 4.40)



Şekil 4.40- İterasyon sayısına bağlı olarak elemanların kesit alanlarında meydana gelen değişim

4.1.25 Adım 25- ANSYS 'den çıkış

1. ANSYS Toolbar da ki “Quit” butonuna basılır.
2. Arzulanan seçenek seçilip “OK” butonuna basılır. [2]

4.2 Problemin Batch Dosyası Hazırlanarak Çözülmesi

Bir önceki bölümde ele alınan üç çubuklu destek sistemi için bu defa batch dosyası oluşturulmak yolu ile yani direk ANSYS komutları kullanılarak optimizasyon analizi gerçekleştirilecektir. [2]

```

/FILNAM, truss
/TITLE, 3 elemanlı bir destek sisteminin optimizasyonu
!      Dizayn değişkeni parametrelerinin tanımlanması
B=1000                      ! Taban uzunluğu
A1=1000                     ! Çubuk-1 ' in kesit alanı
A2=1000                     ! Çubuk-2 ' nin kesit alanı
A3=1000                     ! Çubuk-3 ' ün kesit alanı
!
!      PREP7'ye giriş ve modelin oluşturulması
/PREP7
ET,1,LINK1                  ! 2-D spar (LINK1) elemanı
R,1,A1                      ! Parametrelere göre gerçek sabitler
R,2,A2
R,3,A3
MP,EX,1,2.16e6              ! Young modülü
N,1,-B,0,0                  ! Node 'ların oluşturulması
N,2,0,0,0
N,3,B,0,0
N,4,0,-1000,0
E,1,4                       ! Elemanların oluşturulması
REAL,2
E,2,4
REAL,3
E,3,4
FINISH
!
!      SOLUTION'a giriş ve yüklerin uygulanarak çözümün yapılması
/SOLU
D,1,ALL,0,,3                ! 1,2 ve 3 node 'ları için UX=UY=0
F,4,FX,200000               ! Node-4 teki kuvvetin X bileşeni
F,4,FY,-200000              ! Node-4 teki kuvvetin Y bileşeni
SOLVE
FINISH

```

```

!
!      POST1'e giriş ve dizayn sınırlamalarının (SVs) elde edilmesi.
/POST1
SET, LAST
ETABLE, EVOL, VOLU      ! Her bir eleman hacminin eleman tablosuna
                        ! yerleştirilmesi
SSUM                    ! Eleman tablosundaki verilerin toplanması
*GET, VTOT, SSUM, , ITEM, EVOL ! VTOT=Toplam hacim (parametre elde etme)
RHO=2.85e-4
WT=RHO*VTOT            ! Toplam ağırlığın hesaplanması
ETABLE, SIG, LS, 1      ! Eksenel gerilmelerin eleman tablosuna
                        ! yerleştirilmesi
!      Herbir elemandaki gerilmelerin parametrik olarak tanımlanması
*GET, SIG1, ELEM, 1, ETAB, SIG ! SIG1= Birinci elemandaki eksenel gerilme
*GET, SIG2, ELEM, 2, ETAB, SIG ! SIG2= İkinci elemandaki eksenel gerilme
*GET, SIG3, ELEM, 3, ETAB, SIG ! SIG3= Üçüncü elemandaki eksenel gerilme
!
SIG1=ABS(SIG1)          ! Eksenel gerilmelerin mutlak değerlerinin
                        ! hesaplanması
SIG2=ABS(SIG2)
SIG3=ABS(SIG3)
!
/ESHAPE, 2              ! Elemanların katı olarak gösterilmesi
/VIEW, 1, 1, 1, 1      ! İzometrik görünüş
EPLLOT                  ! Elemanların çizdirilmesi
!
LGWRITE, truss, lgw, , COMMENT ! Veritabanı log dosyasının yazılması
FINISH
!
!      OPTIMIZER'a giriş
/OPT                    ! Optimizer'a giriş (Batch modunda gerekli
                        ! değil)

```

OPANL,truss,lgw	! Analiz dosyasının belirtilmesi(Batch modunda gerekli değil)
!	
OPVAR,B,DV,400,2000	! Dizayn değişkenlerinin tanıtılması
OPVAR,A1,DV,1,1000	
OPVAR,A2,DV,1,1000	
OPVAR,A3,DV,1,1000	
OPVAR,SIG1,SV, ,400	! Dizayn sınırlamalarının tanımlanması
OPVAR,SIG2,SV, ,400	
OPVAR,SIG3,SV, ,400	
!	
OPSAVE,trussvar,opt	! Daha sonra kullanmak üzere optimizasyon veritabanının saklanması.
!	
OPVAR,WT,OBJ, , ,2,	! Hedef fonksiyonun tanımlanması.
!	
OPTYPE,FIRST	! Birinci merteye metodunun kullanılacağını belirtmesi.
OPFRST,45	! Maksimum 45 iterasyon yapılacağını belirtmesi.
OPEXE	! Opt uygulamasının başlatılması.
!	
OPLIST,16	! 16 numaralı en iyi dizayn setinin listelenmesi.
OPLIST,ALL	! Tüm dizayn setlerinin listelenmesi.
!	
/VIEW,1,,,1	! Önden görünüş.
!	
/AXLAB,X,ITERASYON SAYISI	! Ağırlık & İterasyon sayısı grafiğinin çizdirilmesi.
/AXLAB,Y,SISTEMİN AĞIRLIĞI	
PLVAROPT,WT	
!	

/AXLAB,Y,TABAN UZUNLUGU

! B & İterasyon sayısı grafiğinin
çizdirilmesi.

PLVAROPT,B

!

/AXLAB,Y,MAKSIMUM GERİLME

! Maksimum gerilme & İterasyon
sayısı grafiğinin çizdirilmesi.

PLVAROPT,SIG1,SIG2,SIG3

!

/AXLAB,Y,KESIT ALANI

! Kesit alanı & İterasyon sayısı
grafiğinin çizdirilmesi [2].

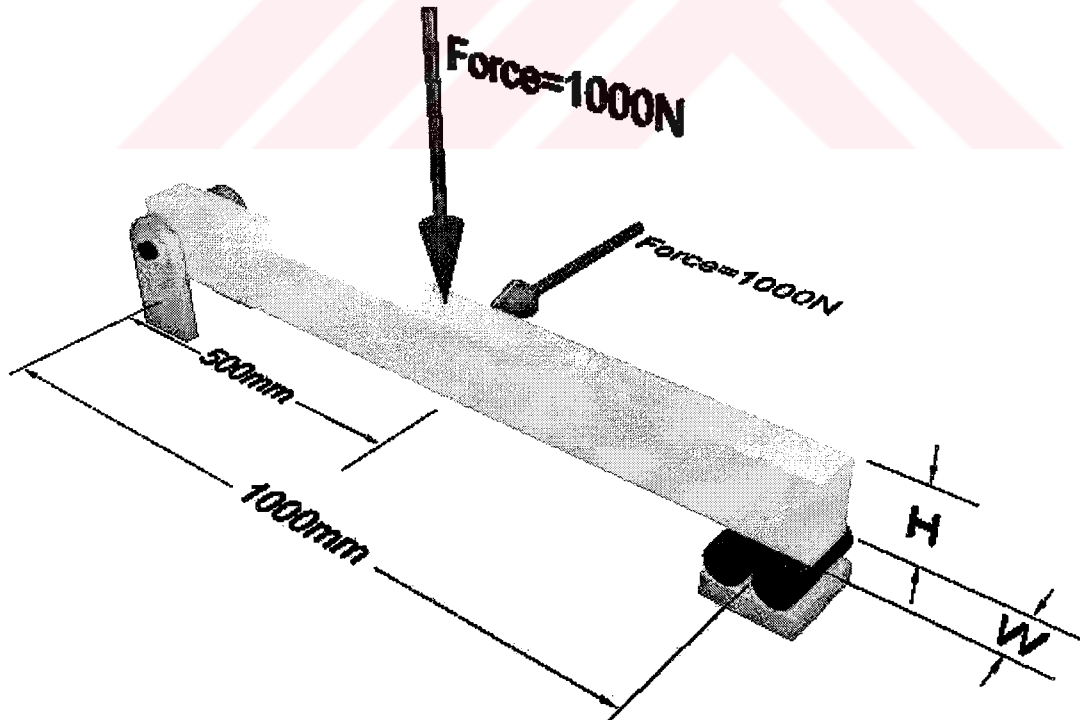


4.3 Uygulama-2

Problemin Tanımı:

Bir çelik kirişin iki farklı yüzeyine şekilde görüldüğü gibi iki adet 1000 N. luk kuvvet etki etmektedir. Optimizasyon işleminde kirişin genişliği (W) ve yüksekliği (H) parametrik olarak tanımlanarak kirişin ağırlığı minimum yapılmaya çalışılacaktır. Kirişin herhangi bir yerinde maksimum gerilme kiriş malzemesinden ötürü 200 MPa değerini aşmamalıdır. Ayrıca W ve H boyutlarının 20 ile 30 mm. arasında olması istenmektedir. Malzemenin Young modülü $E=200000$ MPa dır.

Optimizasyon işlemlerinin tanımlanması için gereken üç önemli unsur, dizayn değişkenleri, dizayn sınırlamaları ve hedef fonksiyonu şunlardır:



Şekil 4.41- Optimizasyonu yapılacak kiriş [5]

Dizayn deęiřkenleri : W ve H boyutları

Dizayn sınırlamaları: $\sigma \leq 200$ MPa

$$20 \leq H \leq 30 \text{ mm.}$$

$$20 \leq W \leq 30 \text{ mm.}$$

Hedef fonksiyonu : Aęırlık

Kiriř için eleman tipi olarak BEAM4 seilecektir. Bu eleman tipi 3 boyutlu analizlerde kullanılabilir ve 6 adet serbestlik derecesi vardır. [5]

Bu problemin analizi için yapılacak iřlemeler menü yolları da verilerek ařaęıda sırasıyla verilmiřtir.  řeklinin bulunduęu satırlar o iřlemlerin bir diyalog kutusunda,  bulunanlar ise grafik ekranda (seim) yapıldıęını gstermektedir.

1. İř adı:

Kiris

2. Bařlık:

File >Change Title...

Bir kirisin optimizasyonu

3. Skalar parametrelerin tanımlanması:

Utility menu >Parameters>Scalar Parameters...



W=20 , H=30

4. Eleman tipinin tanımlanması:

Main Menu >Preprocessor >Element Type >Add/Edit/Delete...



Add... , Beam – 3D elastic 4 (Beam4)

5. Gerek sabitlerin tanımlanması:

Main Menu >Preprocessor >Real Constants...



AREA=W*H , IZZ=W*H*H*H/12 , IYY=H*W*W*W/12 , TKZ=W ,

TKY=H

6. Malzeme özelliğinin tanımlanması:

Main Menu >Preprocessor >Material Props >-Constant-Isotropic...

 OK, EX=200000 Mpa

7. Aktif koordinat sisteminde nodları oluşturma:

Main Menu >Preprocessor >-Modeling-Creat >Nodes >In Active CS...

 N1: 0,0,0 ; N2 : 500,0,0 ; N3 : 1000,0,0

8. Nodlar arasında elemanları oluşturma:

Main Menu >Preprocessor >-Modeling-Creat >Elements >-Auto Numbered- >
Thru Nodes +

 E1: N1→N2 ; E2:N2→N3

9. Eğer çizilen elemanlar ekranda görülmüyorsa, göstermek için;

Utility Menu >Plot >Elements

10. Modelin kaydedilmesi:

Utility Menu >Save as Jobname.db

11. Analiz tipinin belirtilmesi:

Main Menu >Solution >-Analysis Type-New Analysis...

Statik

12. Yerdeğiştirme kısıtlamalarının uygulanması:

Main Menu >Solution >-Loads-Appl >-Structural-Displacement >On Nodes +

 N1  UX,UY,UZ ; 0

 N3  UY,UZ ; 0

13. Kuvvetlerin uygulanması:

Main Menu >Solution >-Loads-Appl >-Structural-Force/Moment >OnNodes +

 N2  FY=-1000 N

 N2  FZ = 1000 N

14. Çözümün başlatılması:

Main Menu >Solution >-Solve-Current LS

 OK

15. Eleman hacimlerinin eleman tablosuna yazılması:

General Postprocessor >Element Table >Define Table....

 OK, User Label = EVOL (Eleman tablosu etiketi)

İtem = Geometry, VOLU (Tabloya yazılacak veriler: hacimler)

16. Eleman tablosuna yazılan hacimlerin toplanması:

General Postprocessor >Element Table >Sum of Each Item....

Sonuç olarak 600000 mm³ çıkacaktır.

17. Bulunan toplam hacmin skalar parametre olarak atanması:

Utility Menu > Parameters > Get Scalar Data....

 Results data, Elem table sums

 Parametre ismi: VSUM

18. Gerilmeler için eleman tablosunun yazılması:

General Postprocessor >Element Table >Define Table....

 OK, User Label = STRESS (Eleman tablosu etiketi)

İtem = By sequence number, NMISC,1 (Tabloya yazılacak veriler: gerilmeler)

19. Gerilme eleman tablosundaki değerleri büyükten küçüğe doğru sıralama:

General Postprocessor >List Results >Sort Elems....

 Sort elements based on = STRESS

20. Sıralanan gerilmelerinin en büyük olanını skalar parametre olarak atamak:

Utility Menu > Parameters > Get Scalar Data....

 Results data, Other operations

 Parametre ismi = STR

From sort oper'n, Maximum value

29. Oluşan dizayn setlerinin görüntülenmesi:

Main Menu >Design Opt >-Design sets-List...

 All

Oluşan bütün dizayn setleri Tablo 4.3 'de verilmiştir. En iyi dizayn seti olarak 28. set görülmektedir. Bu setteki parametre değerleri şöyledir.

STR = 199.89 MPa

H = 24.831 mm

W = 24.504 mm

VSUM = 608450 mm³



Tablo 4.3- Uygulama-2 için oluşan bütün dizayn setleri

		SET 1	SET 2	SET 3	SET 4	SET 5
		(INFEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)
STR	(SV) >	208.33	198.68	199.10	199.19	199.30
H	(DV)	30.000	30.000	29.950	29.919	29.915
W	(DV)	20.000	20.603	20.607	20.620	20.615
VSUM	(OBJ)	.60000E+06	.61810E+06	.61717E+06	.61692E+06	.61668E+06
		SET 6	SET 7	SET 8	SET 9	SET 10
		(FEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)
STR	(SV)	199.42	199.53	199.68	199.54	199.39
H	(DV)	29.911	29.907	29.897	29.900	29.902
W	(DV)	20.610	20.604	20.601	20.608	20.617
VSUM	(OBJ)	.61645E+06	.61622E+06	.61591E+06	.61619E+06	.61648E+06
		SET 11	SET 12	SET 13	SET 14	SET 15
		(FEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)
STR	(SV)	199.64	199.53	199.40	199.61	199.45
H	(DV)	29.888	29.890	29.892	29.818	29.788
W	(DV)	20.609	20.615	20.622	20.654	20.683
VSUM	(OBJ)	.61596E+06	.61618E+06	.61643E+06	.61586E+06	.61610E+06
		SET 16	SET 17	SET 18	SET 19	SET 20
		(FEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)
STR	(SV)	199.15	199.58	198.42	199.57	199.65
H	(DV)	23.739	23.677	24.554	25.186	25.076
W	(DV)	25.708	25.740	24.902	24.187	24.286
VSUM	(OBJ)	.61027E+06	.60943E+06	.61146E+06	.60918E+06	.60898E+06
		SET 21	SET 22	SET 23	SET 24	SET 25
		(FEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)	(FEASIBLE)
STR	(SV)	199.63	199.48	199.33	199.21	199.55
H	(DV)	24.843	24.849	24.855	24.859	24.845
W	(DV)	24.513	24.519	24.526	24.532	24.518
VSUM	(OBJ)	.60897E+06	.60928E+06	.60959E+06	.60984E+06	.60915E+06
		SET 26	SET 27	SET 28		
		(FEASIBLE)	(FEASIBLE)	(FEASIBLE)		
STR	(SV)	199.68	199.81	199.89		
H	(DV)	24.839	24.834	24.831		
W	(DV)	24.513	24.507	24.504		
VSUM	(OBJ)	.60888E+06	.60862E+06	.60845E+06		

4.4 Uygulama-3

Bölüm 3.3.1 'de dizayn değişkenlerinin seçiminde dikkat edilecek bazı hususlar açıklanmıştı. Konik veya eğrisel yapıda bir şeklin (Şekil 3.2.(d)) oluşmasına izin verecek şekilde dizayn değişkenlerinin tanımlanması, aynı sistem için daha küçük ağırlıkla tasarıma imkan veriyordu.

Bu uygulamada yukarıda bahsedilen şekilde parametrik olarak oluşturulmuş bir ankastre kirişin optimizasyonu yapılacaktır.

Problemin tanımı:

Aşağıdaki şekilde görülen ankastre kiriş sisteminin bir ucuna M momenti etki etmektedir. Kirişin herhangi bir noktasında maksimum gerilme $\sigma_{\max}$ değerini, maksimum deformasyon da $\delta_{\max}$ değerini aşmamalıdır. Kirişin kalınlığı, boyunca değişebilir. Fakat uç noktasındaki kalınlığı t kadar olmalıdır. [2]

$$E = 10.10^6 \text{ psi } (=6.9.10^{10} \text{ Pa})$$

$$l = 10 \text{ in } (25.4 \text{ mm})$$

$$\sigma_{\max} = 30000 \text{ psi}$$

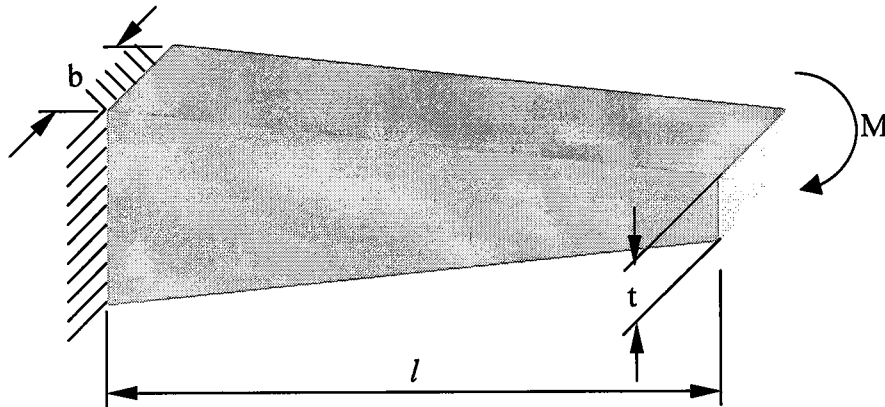
$$\nu = 0.3$$

$$b = 1 \text{ in } (2.54 \text{ mm})$$

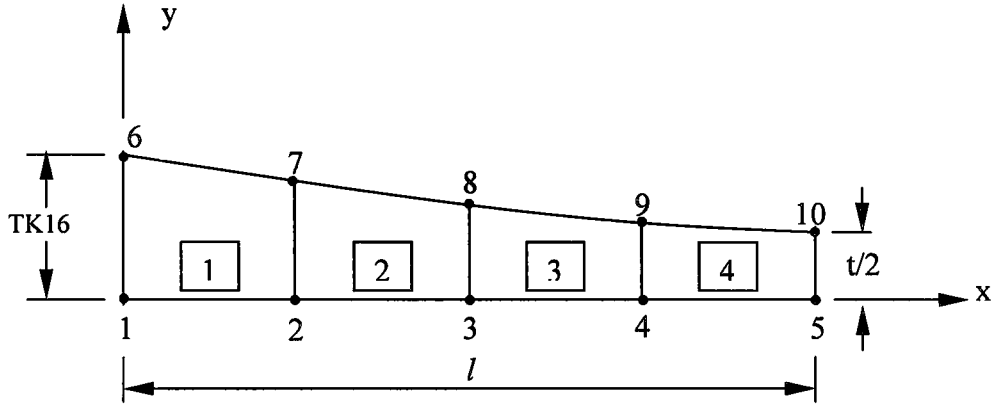
$$\delta_{\max} = 0.5 \text{ in}$$

$$M = 450 \text{ lb-in } (=1,13.10^{-1} \text{ N.m})$$

$$t = 0.3 \text{ in } (0.762 \text{ mm})$$



Şekil 4.42- Optimizasyonu yapılacak ankastre kiriş [2]



Şekil 4.43- Anahtar noktaları (keypoints), alanları, ve TK16 isimli dizayn değişkenini gösteren yarı simetrik katı model [2]

Yukarıdaki şekilde görünen 6, 7, 8, 9, 10 numaralı anahtar noktalar bir spline ile birleştirilmişlerdir. Kirişin simetrisinden dolayı işlemlerde basitlik sağlaması için yarısı modellenmiştir. Bu yüzden hedef fonksiyonu için toplam eleman hacmi iki ile çarpılacaktır. Maksimum gerilme ve maksimum deformasyon değerleri SVs olarak tanımlanacaktır.

Bunlara ek olarak tanımlanan anahtar noktalar arasına dört adet dizayn değişkeni tanımlanmıştır. Bunlardan birisi Şekil 4.43 'de de gösterilen ve 1 ile 6 numaralı anahtar noktalara arasına tanımlanan TK16 dır. Diğer dizayn değişkenleri ise yine tanımlanmalarında kullanılan anahtar noktalara göre, TK27, TK38, TK49 dur.

Problem hem alt problem yaklaşımı metodu hem de birinci mertebe metodu ile çözülmüştür. Her ikisinde de aynı başlangıç dizaynı kullanılmıştır. Alt problem yaklaşımı metodunda soldan sağa doğru bir eğim temin etmek için üç adet geometrik SVs tanımlanmıştır. Bunlar DIF1, DIF2, DIF3 tür. Bu geometrik sınırlamalar, birinci mertebe metodunda kullanılan algoritmanın sağlamlığını göstermek için kullanılmamıştır. [2]

Bu problemin batch modundaki çözümü aşağıda verilmiştir.

! Parametrelere başlangıç değerlerinin atanması

TK16=0.25

TK27=0.25

TK38=0.25

TK49=0.25

!

! PREPROCESSOR'e giriş

/PREP7

smrt,off

/TITLE, Bir ankastre kirisin optimizasyonu

ET,1,PLANE42

MP,EX,1,10E6

MP,NUXY,1,0.3

K,1

K,5,10

K,6,,TK16

! Dizayn değişkenlerine göre anahtar noktaların tanımı

K,7,2.5,TK27

K,8,5,TK38

K,9,7.5,TK49

K,10,10,0.15

SPLINE,6,7,8,9,10

! Üst kenarın bir SPLINE ile çizdirilmesi

L,1,6

*REPEAT,5,1,1

LSEL,S,LINE,,5,9

LESIZE,ALL,,1

LSEL,ALL

A,1,2,7,6,

*REPEAT,4,1,1,1,1

ESIZE,,4

AMESH,ALL

NSEL,S,LOC,Y

DSYM,SYMM,X

NSEL,S,LOC,X

```

DSYM,ASYM,Y
NSEL,ALL
FK,10,FX,1500          ! Kuvvetin uygulanması
DK,1,ALL,0
FINISH
!      SOLUTION modülüne giriş
/SOLU
SOLVE
FINISH
!      POSTPROCESSOR'e giriş
/POST1
SET,LAST
ETABLE,VOLU,VOLU
PRNSOL,S,PRIN
NSORT,S,1              ! Maksimum gerilmenin elde edilmesi
NSEL,S,LOC,X,0,9      ! Yük noktasındaki yerel gerilmeleri göz ardı et
*GET,STRS,SORT,,MAX
NSEL,ALL
SSUM
*GET,TVOL,SSUM,,ITEM,VOLU  ! TVOL=Toplam eleman hacmi
TVOL=TVOL*2
NSEL,S,LOC,X,9.9,10.1
PRNSOL,U,Y
NSORT,U,Y,,1
PRNSOL,U,Y
*GET,DEFL,SORT,,MAX    ! Maksimum deformasyonun elde edilmesi
*STATUS
DEFL=ABS(DEFL)
! Soldan sağa bir eğim oluşmasından emin olmak için SVs tanımlanması
DIF1=TK16-TK27
DIF2=TK27-TK38
DIF3=TK38-TK49
FINISH

```

! OPT modülüne giriş

/OPT

OPVAR,TVOL,OBJ,,0.1 ! Hedef fonksiyonunun tanımlanması

OPVAR,STRS,SV,,30000 ! SVs tanımlanması

OPVAR,DEFL,SV,,0.50

OPVAR,DIF1,SV,0,0.1

OPVAR,DIF2,SV,0,0.1

OPVAR,DIF3,SV,0,0.1

OPVAR,TK16,DV,0.15,0.27,0.001 ! Dizayn değişkenleri

OPVAR,TK27,DV,0.15,0.27,0.001

OPVAR,TK38,DV,0.15,0.27,0.001

OPVAR,TK49,DV,0.15,0.27,0.001

OPSAVE,INITIAL,OPT ! Optimizasyon veritabanının kaydedilmesi

OPTYPE, SUBP ! Alt problem yaklaşımı metodu

OPSUBP,30 ! En fazla 30 iterasyon

OPEXE

OPLIST,ALL,,1 ! Dizayn setlerinin listelenmesi

/AXLAB,Y,VOLUME(TVOL)

PLVAROPT,TVOL

OPRESU,INITIAL,OPT

OPVAR,DIF1,DEL

OPVAR,DIF2,DEL

OPVAR,DIF3,DEL

OPTYPE,FIRST

OPFRST,20

STATUS

OPEXE

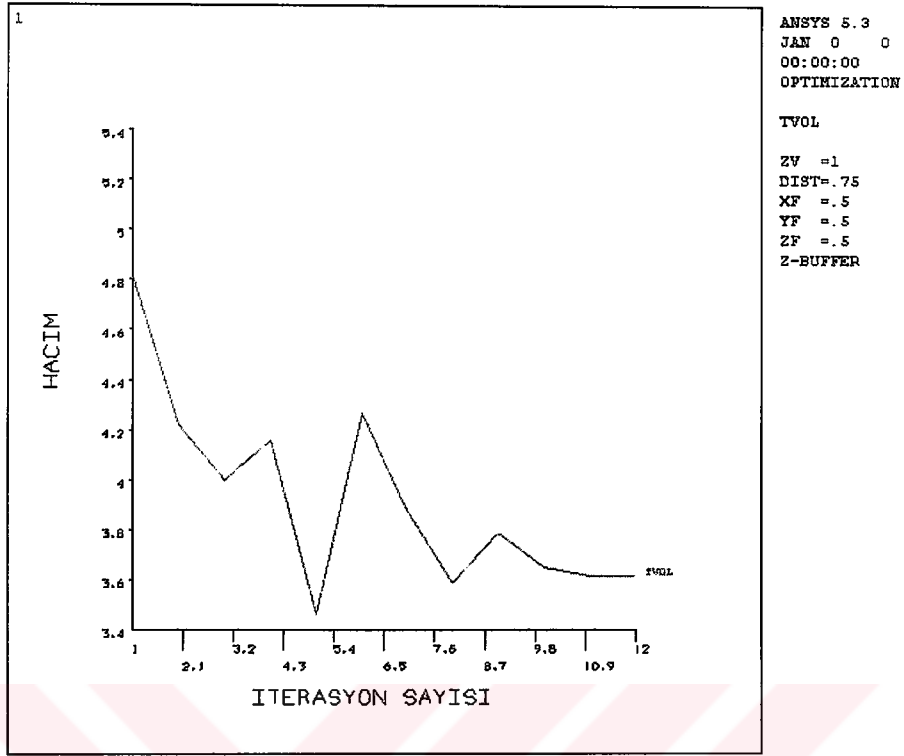
OPLIST,ALL,,1

/AXLAB,Y,VOLUME(TVOL)

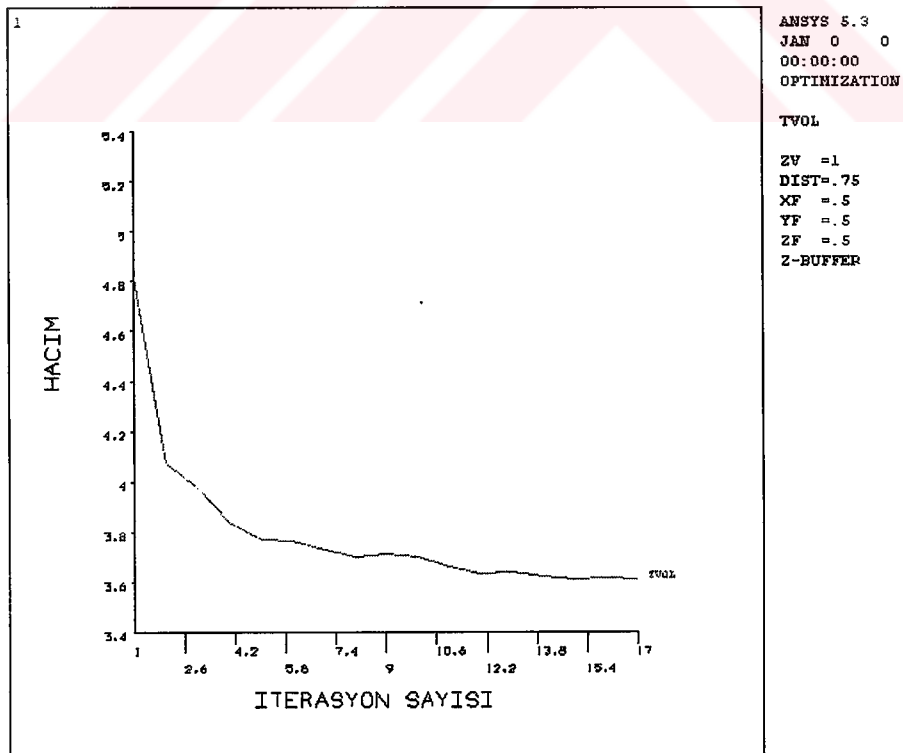
PLVAROPT,TVOL

FINISH

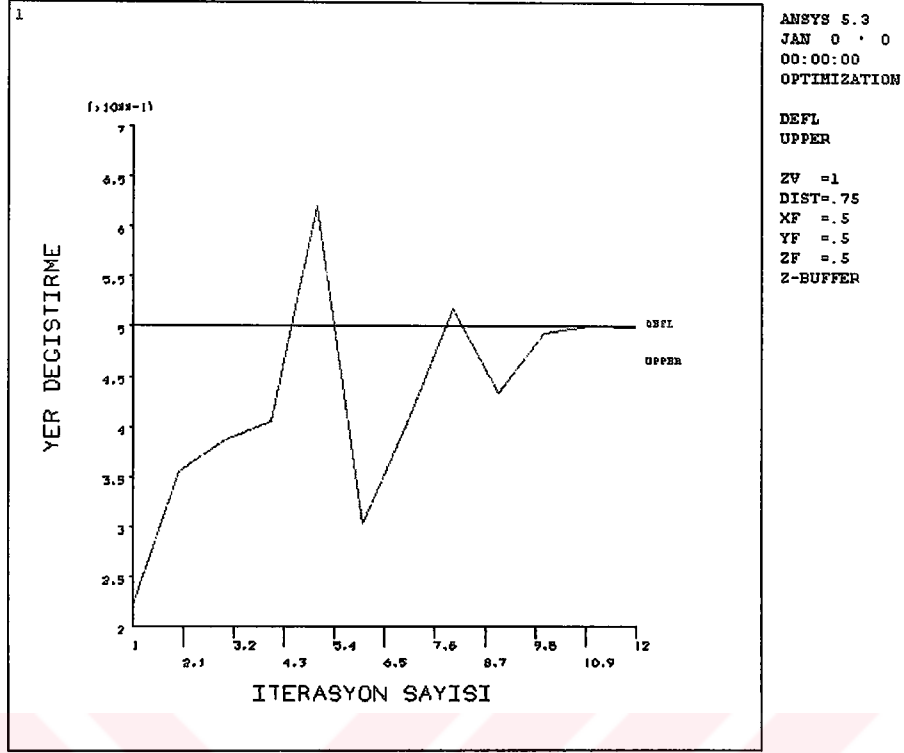
[2]



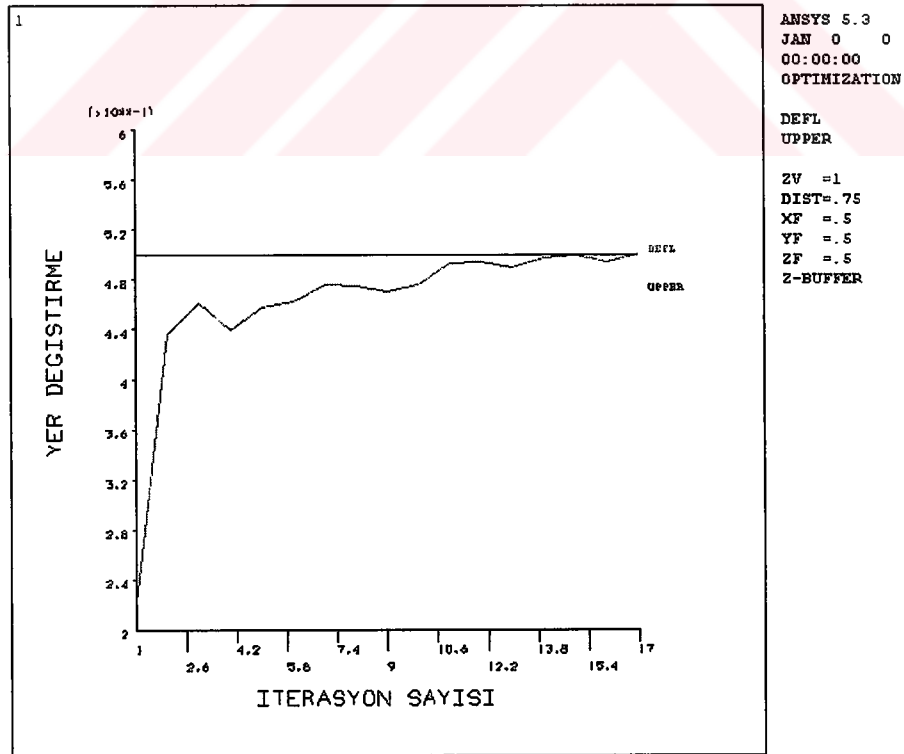
Şekil 4.44- Toplam hacmin iterasyon sayısı ile değişimi (altproblem yaklaşımı metodu)



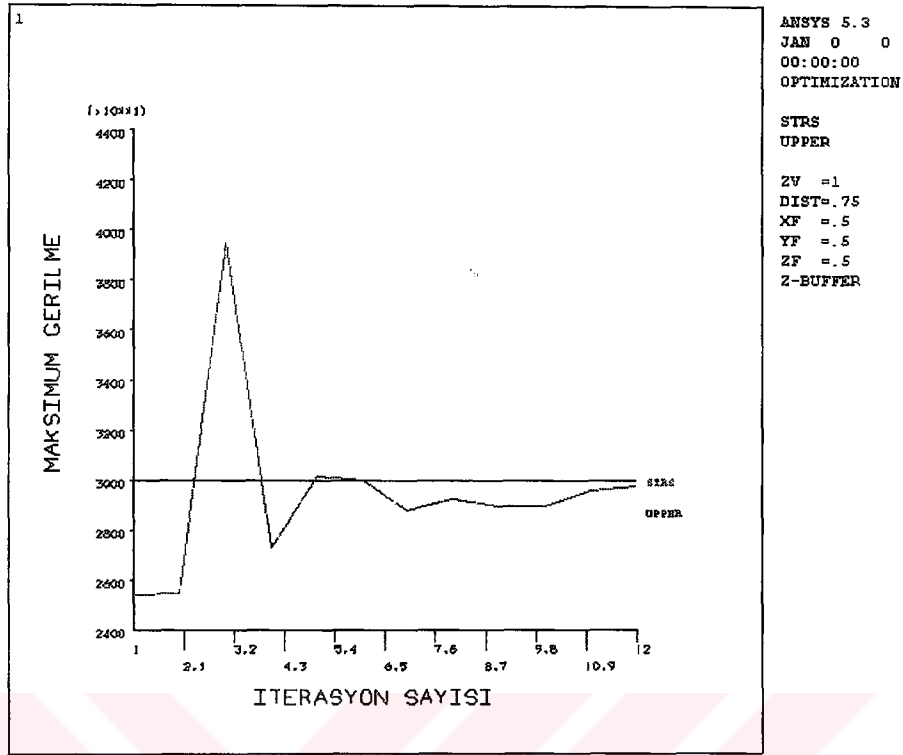
Şekil 4.45- Toplam hacmin iterasyon sayısı ile değişimi (birinci mertebe metodu)



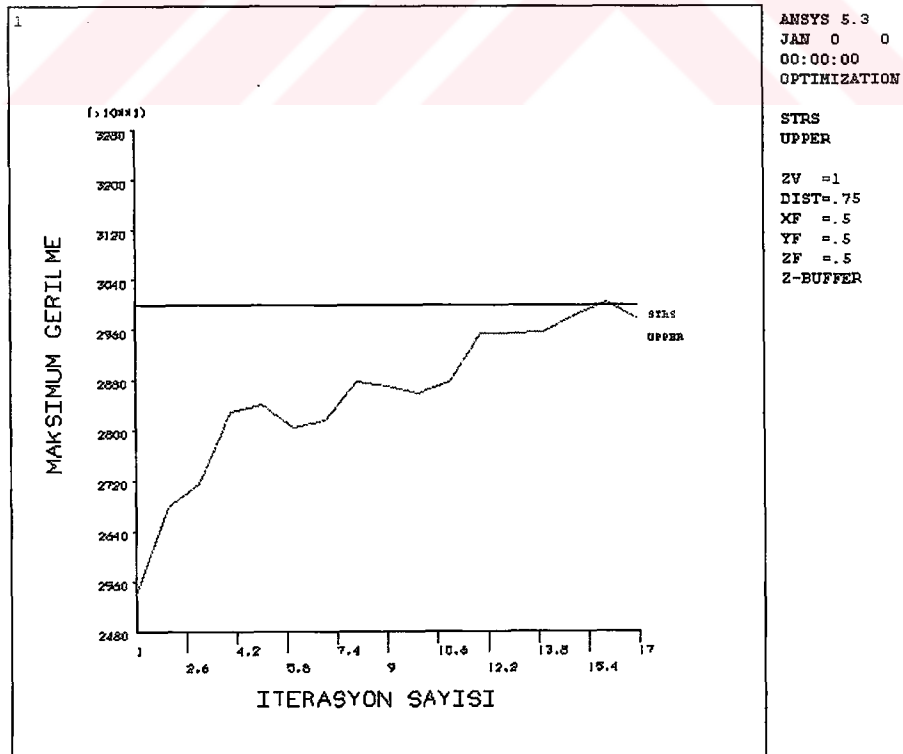
Şekil 4.46- Kirişte oluşan deformasyonun iterasyon sayısı ile değişimi (altproblem yaklaşımları metodu)



Şekil 4.47- Kirişte oluşan deformasyonun iterasyon sayısı ile değişimi (birinci mertebe metodu)



Şekil 4.48- Maksimum gerilmenin iterasyon sayısı ile değişimi (altproblem yaklaşımı metodu)



Şekil 4.49- Maksimum gerilmenin iterasyon sayısı ile değişimi (birinci mertebe metodu)

BÖLÜM 5

ANSYS PROGRAMINDA KULLANILAN OPTİMİZASYON TEKNİKLERİ

5.1 Giriş

Bir bilgisayar programının kullandığı algoritmayı anlamak her zaman faydalı olacaktır. ANSYS programında kullanılan optimizasyon teknikleri “optimizasyon metotları” ve “optimizasyon araçları” olmak üzere iki alt başlıkta incelenebilir.

Optimizasyon metotları, belli sınırlamalara göre, verilen bir fonksiyonu (hedef fonksiyonu) minimum yapmaya çalışan geleneksel tekniklerdir. Bu metotlar Bölüm 2’de de bahsi geçen sayısal yöntemlere benzemektedir. ANSYS programında “Altproblem yaklaşımı” (Subproblem approximation) metodu ve “Birinci merteye” (First order) metodu olmak üzere iki farklı metot kullanılmaktadır. Bunlara ek olarak, kullanıcı kendi bildiği bir optimizasyon algoritması da oluşturabilir. Bu durumda ANSYS algoritması kullanılmaz. Bu metotlardan birisini kullanmak için, bir hedef fonksiyonunun tanımlanması şarttır. [2]

- *Altproblem yaklaşımı (Subproblem Approximation) metodu*: Sıfırıncı mertebeden (türevsel bilgiler içermeyen) olan ve tüm bağımlı değişkenlere (SVs ve hedef fonksiyonu) yaklaşımlar kullanan bir metottur. Çok geniş sahadaki mühendislik problemlerine uygulanabilen genel bir metottur.
- *Birinci merteye (First Order) metodu*: Bu metot, dizayn değişkenlerine göre bağımlı değişkenlerin eğimlerini içeren türevsel bilgileri kullanır. Eğimler birinci mertebeden türevleri ifade ettiklerinden, bu metot birinci merteye olarak isimlendirilir. Bağımlı değişkenlerin çok geniş bir aralıkta sınırlandırıldığı problemler için çok yüksek doğrulukta işlem yapar.

- *Kullanıcı kaynaklı (User Defined) metot:* ANSYS optimizier mantığı yerine kullanıcı tarafından tanımlanan bir algoritmayı kullanılabilen optimizasyon rutinidir.

Optimizasyon araçları, problemin dizayn uzayını anlamak ve ölçmek için kullanılan tekniklerdir. Bunda minimizasyon bir hedef olmayabileceğinden, bu araçların kullanılması için hedef fonksiyonu gerekli değildir. Fakat dizayn değişkenleri tanımlanmalıdır. Kullanılan araçlar şunlardır: [2]

- *Tek döngü çalıştırma (Single Loop Run):* Bu araçla sadece bir döngü gerçekleştirilir ve sonlu elemanlar (FEA) çözümü oluşturulur. Amaç dizayn değişkenlerinin seçilen bazı değerleri için neticenin ne olduğunu öğrenmektir. Birden fazla durumda inceleme yapılacaksa her bir döngü çalıştırılmadan önce dizayn değişkenlerinin değerleri değiştirilir.
- *Rasgele dizayn oluşturma (Random Design Generation):* Her bir döngüde rasgele dizayn değişkeni değerleri için, birçok döngü yapılır. Yapılacak döngülerin maksimum sayısı ve arzulanan sayıda uygun seti veren döngü sayısı belirtilebilir. Bu araç, dizayn uzayı ve yapılacak analiz için uygun dizayn setleri hakkında geniş kapsamlı bilgi elde etmek için kullanışlıdır.
- *Süpürme (Sweep Generation):* Referans bir dizayn setinden başlayarak birkaç dizayn seti oluşturur. Bunu yaparken bir dizayn değişkenini alıp, bunun değerini belirtilen sınırlar içinde değiştirir.
- *Faktöriyel değerlendirme (Factorial Evaluation):* Bu araç dizayn değişkenlerinin tüm uç noktalarında dizayn setleri oluşturmak için kullanılır. Bu teknik, “deney dizaynı” olarak bilinen, tüm ve parça faktöriyel analizi olarak iki seviye kullanan teknoloji ile ilişkilidir.
- *Eğim değerlendirme (Gradient Evaluation):* Kullanıcı tarafından belirtilmiş bir dizayn setinde, DVs’a göre SVs ve hedef fonksiyonunun eğimlerini hesaplar. Bu araç kullanarak yerel dizayn hassasiyetleri de araştırılabilir.

5.2 Optimizasyon Metotları

Dizayn değişkenleri vektörü şöyle gösterilir:

$$\mathbf{x} = [x_1 \ x_2 \ x_3 \ \dots \ x_n]^T \quad (5.1)$$

dizayn değişkeni sınırlamaları ise

$$\underline{x}_i \leq x_i \leq \bar{x}_i \quad (i=1,2,3,\dots,n) \quad (5.2)$$

burada,

n : Dizayn değişkenlerinin sayısı.

$$f=f(\mathbf{x}) \quad (5.3)$$

$$g_i(\mathbf{x}) \leq \bar{g}_i \quad (i = 1,2,3,\dots,m_1) \quad (5.4)$$

$$\underline{h}_i \leq h_i(\mathbf{x}) \quad (i = 1,2,3,\dots,m_2) \quad (5.5)$$

$$\underline{w}_i \leq w_i(\mathbf{x}) \leq \bar{w}_i \quad (i = 1,2,3,\dots,m_3) \quad (5.6)$$

f : Hedef fonksiyonu

g_i, h_i, w_i : Alt ve üst çizgili ifadeleri, alt ve üst sınırlarını gösteren sınırlamalar.

$m_1 + m_2 + m_3$: SVs sayısı

Uygun (feasible) dizayn uzayı belirtilirken her bir SV ye bir tolerans eklenir. Dizayn seti vektörü ve SV sınırlamaları şöyle gösterilir:

$$\mathbf{x}^* = [x_1^* \ x_2^* \ x_3^* \ \dots \ x_n^*]^T \quad (5.7)$$

sadece aşağıdaki durumlar sağlanıyorsa dizayn uygundur.

$$g_i^* = g_i(\mathbf{x}^*) \leq \bar{g}_i + \alpha_i \quad (i = 1,2,3,\dots,m_1) \quad (5.8)$$

$$\underline{h}_i - \beta_i \leq h_i^* = h_i(\mathbf{x}^*) \quad (i = 1,2,3,\dots,m_2) \quad (5.9)$$

$$\underline{w}_i - \gamma_i \leq w_i^* = w_i(x^*) \leq \bar{w}_i + \gamma_i \quad (i = 1, 2, 3, \dots, m_3) \quad (5.10)$$

burada;

α_i, β_i ve γ_i toleransları ifade etmektedir

ve bunlara ek olarak dizayn değişkeni sınırlamalarına tolerans eklenmediği için;

$$\underline{x}_i \leq x_i^* \leq \bar{x}_i \quad (i = 1, 2, 3, \dots, n) \quad (5.11)$$

5.2.1 Altproblem yaklaşımı (Subproblem approximation) metodu

Altproblem yaklaşımı metodu sıfıncı mertebeden, yani sadece bağımsız değişkenleri gerektiren fakat türevlerini gerektirmeyen bir metottur. Bu metotta iki önemli konsept vardır. Hedef fonksiyonu ve dizayn sınırlamaları (SVs) için yaklaşımların (approximations) kullanımı, ve sınırlanmış (constrained) optimizasyon probleminin, bir sınırlanmamış (unconstrained) probleme dönüştürülmesi. İlk olarak bağımlı değişkenler en küçük kareler uygunluğu kullanılarak, yaklaşımlarla değiştirilir ve sınırlanmış dizayn optimizasyonu problemi, penaltı fonksiyonları yardımı ile sınırlanmamış dizayn optimizasyonu problemine dönüştürülür. Daha sonra altproblem olarak adlandırılan bu yaklaşımlı penaltı fonksiyonu kullanılarak minimize etme işlemi başlatılır. Bu metot için her iterasyon bir tam analiz döngüsüne eşittir. Bu metod OPTYPE,SUBP komutu ile seçilir.

Metot yaklaşımlarla fonksiyonları oluşturduğu için ön bilgi olarak belirli sayıda dizayn seti bulunmalıdır. Bu ön bilgiler kullanıcı tarafından optimizasyon araçları veya metotları kullanılarak oluşturulabilir. Eğer oluşturulmazsa metot kendisi rasgele dizayn setleri oluşturur ve bunları kullanarak yaklaşım fonksiyonlarını oluşturur. [2]

5.2.1.1 Fonksiyon yaklaşımları

(5.3) $\Rightarrow$ (5.6) denklemleri ile ifade edilmiş olan problemin minimizasyon işlemi için ilk olarak bütün bağımlı değişkenler (hedef fonksiyonu ve SVs), " ^ " işareti ile yaklaşımlar olarak temsil edilir. Hedef fonksiyonu için;

$$\hat{f}(x) = f(x) + hata \quad (5.12)$$

ve benzer şekilde sınırlamalar için;

$$\hat{g}(x) = g(x) + hata \quad (5.13)$$

$$\hat{h}(x) = h(x) + hata \quad (5.14)$$

$$\hat{w}(x) = w(x) + hata \quad (5.15)$$

Yaklaşımlar için olabilecek en karmaşık form ikinci derecedendir. Hedef fonksiyonunun örneğini kullanarak;

$$\hat{f} = a_0 + \sum_i^n a_i . x_i + \sum_i^n \sum_j^n b_{ij} . x_i . x_j \quad (5.16)$$

Her bir uyguluk (fit) program tarafından belirlenir ve iterasyondan iterasyona değişir. Denklem (5.16) da ki a_i ve b_{ij} katsayıları, bir ağırlıklı en küçük kareler tekniği kullanılarak belirlenir. Örneğin, hedef fonksiyonu için ağırlıklı en küçük kareler hata normu aşağıdaki gibidir.

$$E^2 = \sum_{j=1}^{n_d} \phi^{(j)} (f^{(j)} - \hat{f}^{(j)})^2 \quad (5.17)$$

burada,

$\phi^{(j)}$: j . dizayn seti ile ilgili ağırlık.

n_d : Dizayn setlerinin o andaki sayısını göstermektedir.

Her bir SV için benzer E^2 normları oluşturulur. E^2 katsayılarına göre minimize edilerek, denklem (5.17) de ki katsayılar belirlenir. Yukarıda kullanılan ağırlıklar aşağıdaki yollardan birisi ile hesaplanır.

1. Hedef fonksiyonunun değerleri baz alınarak. Küçük hedef fonksiyonu değerlerine sahip olan dizayn setleri yüksek ağırlığa sahiptir.
2. Dizayn değişkenlerinin değerleri baz alınarak. En iyi dizayna yakın olan dizayn setleri, yüksek ağırlık alırlar.
3. Fizibilite baz alınarak. Uygun setler yüksek, uygun olmayan setler düşük ağırlığa sahiptirler.
4. Yukarıda tarif edilen üç ağırlığın bir kombinezonu baz alınarak.
5. Tüm ağırlıklar birim büyüklükte alınarak ; bütün j 'ler için $\phi^{(j)} = 1$

Yaklaşımları oluşturabilmek için belirli sayıda dizayn seti var olmalıdır. Eğer yoksa aşağıda belirtilen şartlar altında, gerekli sayıya ulaşılan kadar bu dizayn setleri rasgele oluşturulur.

$n_d < n + 2 \rightarrow$ ise rasgele dizayn setleri oluşturulur.

$n_d \geq n + 2 \rightarrow$ ise yaklaşımlar oluşturulur. (5.18)

burada,

n : Dizayn değişkenlerinin sayısı

n_d : Dizayn setlerinin sayısı

Eğer daha fazla veri (dizayn setleri) olursa, denklem (5.16) da ki terimlerin sayısı da artacaktır. [2]

5.2.1.2 Altproblem yaklaşımının minimum yapılması

Fonksiyon yaklaşımlarının kullanılması ile minimizasyon problemi aşağıdaki şekli alır.

Hedef fonksiyonu:

$$\hat{f} = \hat{f}(x) \quad (5.19)$$

Sınırlamalar:

$$\underline{x}_i \leq x_i \leq \bar{x}_i \quad (i=1,2,3,...,n) \quad (5.20)$$

$$\hat{g}_i(x) \leq \bar{g}_i + \alpha_i \quad (i=1,2,3,...,m_1) \quad (5.21)$$

$$\underline{h}_i - \beta_i \leq \hat{h}_i(x) \quad (i=1,2,3,...,m_2) \quad (5.22)$$

$$\underline{w}_i - \gamma_i \leq \hat{w}_i(x) \leq \bar{w}_i + \gamma_i \quad (i=1,2,3,...,m_3) \quad (5.23)$$

Bundan sonraki adım (5.19) dan (5.23) e kadar olan denklemlerin, sınırlanmış bir problemten, sınırlanmamış bir probleme dönüştürülmesidir. Bu işlem, Bölüm 2’de de bahsi geçen, penaltı fonksiyonları yardımı ile yapılır ve problem sadece aşağıdaki problemin minimize edilmesi halini alır.

$$F(x, p_k) = \hat{f} + f_0 \cdot p_k \left(\sum_{i=1}^n X(x_i) + \sum_{i=1}^{m_1} G(\hat{g}_i) + \sum_{i=1}^{m_2} H(\hat{h}_i) + \sum_{i=1}^{m_3} W(\hat{w}_i) \right) \quad (5.24)$$

Bu formülde X, dizayn değişkenleri sınırlamaları için kullanılan penaltı fonksiyonu, G, H ve W diğer sınırlamalar için kullanılan penaltı fonksiyonlarını göstermektedir. Referans hedef fonksiyonu değeri olan f_0 , birimlerin birbirine uyması için kullanılmıştır. Sınırlanmamış hedef fonksiyonu olan ve aynı zamanda cevap yüzeyi olarak da adlandırılan $F(x, p_k)$ nın, dizayn değişkenleri ve bir cevap yüzeyi parametresi olan p_k ile değiştiği görülmektedir. (5.24) denklemini çözmek için bir "ardışık sınırlanmamış minimizasyon tekniği" (SUMT: Sequential unconstrained minimization technique) kullanılır. k indisi, alt problemin çözümü esnasında kullanılan alt iterasyonların sayısını göstermektedir ki bunun yardımı ile daha yaklaşık ve kesin sonuçların elde edilmesi için p_k parametresinin değeri artar.

Oluşturulan bütün penaltı fonksiyonları genişletilmiş iç penaltı fonksiyonu tipindedir. Örneğin; üst sınır civarında dizayn değişkeni penaltı fonksiyonu şu formdadır:

$$X(x_i) = \begin{cases} c_1 + c_2 / (\bar{x} - x_i) & \text{eğer } x_i < \bar{x} - \varepsilon(\bar{x} - \underline{x}) \text{ ise} \\ c_3 + c_4 \cdot (x_i - \bar{x}) & \text{eğer } x_i \geq \bar{x} - \varepsilon(\bar{x} - \underline{x}) \text{ ise} \end{cases} \quad i=(1,2,3,...,n) \quad (5.25)$$

burada,

c_1, c_2, c_3, c_4 : içeriden hesaplanan sabitler
 ε : çok küçük bir pozitif sayı

Diğer sınırlamalar da yine üst sınır civarında aşağıdaki formdadır:

$$W(w_i) = \begin{cases} d_1 + d_2 / (\bar{w}_i - \hat{w}_i) & \text{eğer } \hat{w}_i < \bar{w}_i - \varepsilon(\bar{w}_i - \underline{w}_i) \text{ ise} \\ d_3 + d_4 (\hat{w}_i - \bar{w}_i) & \text{eğer } \hat{w}_i \geq \bar{w}_i - \varepsilon(\bar{w}_i - \underline{w}_i) \text{ ise} \end{cases} \quad (i=1,2,3,...,m_1) \quad (5.26)$$

burada,

d_1, d_2, d_3, d_4 : içeriden hesaplanan sabitler

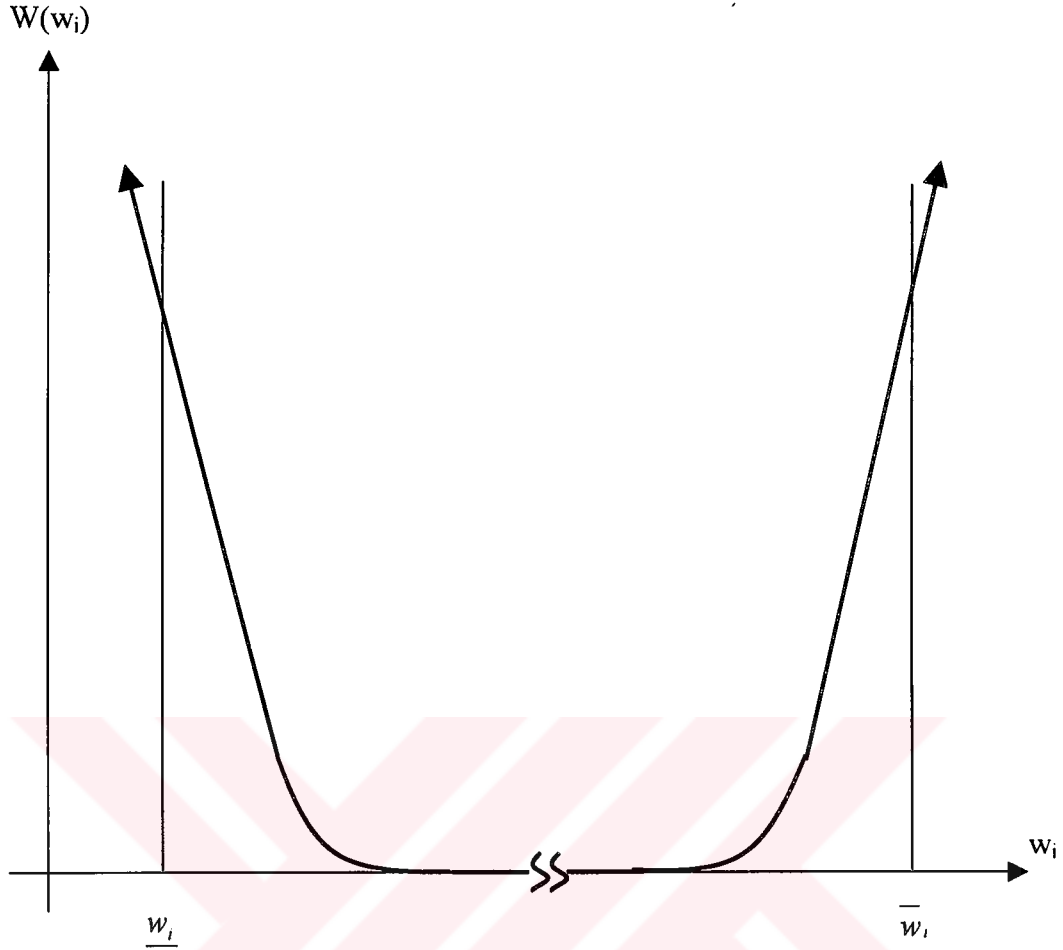
G ve H için de penaltı fonksiyonu yukarıdaki formüle benzer şekilde yazılabilir. Sonuç Şekil 5.1 'de görülmektedir.

SUMT algoritması kullanılarak j. iterasyondaki sınırlanmamış minimum hedef fonksiyonuna ($\tilde{F}^{(j)}$) ulaşılır.

$$x^{(j)} \rightarrow \tilde{x}^{(j)} \quad F^{(j)} \rightarrow \tilde{F}^{(j)} \quad (5.27)$$

burada,

$\tilde{x}^{(j)}$: $\tilde{F}^{(j)}$ 'ye karşılık gelen dizayn değişkeni vektörü.



Şekil 5.1- SVs uzayında genişletilmiş iç penaltı fonksiyonu

Her bir iterasyonun en son adımı bir sonraki iterasyonda ($j+1$) kullanılacak olan dizayn değişkeni vektörünün belirlenmesidir. $x^{(j+1)}$ vektörü aşağıdaki denkleme göre belirlenir.

$$x^{(j+1)} = x^{(b)} + C(\tilde{x}^{(j)} - x^{(b)}) \quad (5.28)$$

burada,

$x^{(b)}$: en iyi dizayn değişkeni seti sabitleri

C : uygun olmayan çözümlerin sayısına bağlı olarak içeriden seçilen ve 0.0 ile 1.0 arasında değişen sayı. [2]

5.2.1.3 Yaklaşım

Altproblem yaklaşımı iterasyonları, yaklaşım sağlanana veya belirtilen durum oluştuğunda kesilene kadar devam eder. Bu iki durum, denklem (5.18) e göre kontrol edilebilir.

Aktif dizayn seti ($x^{(j)}$) veya bir önceki dizayn seti ($x^{(j-1)}$) veya en iyi dizayn setinden ($x^{(b)}$) herhangi biri uygun ise ve aşağıdaki durumlardan birisi sağlanıyorsa yaklaşım sağlanmış olarak kabul edilir.

$$|f^{(j)} - f^{(j-1)}| \leq \tau \quad (5.29)$$

$$|f^{(j)} - f^{(b)}| \leq \tau \quad (5.30)$$

$$|x_i^{(j)} - x_i^{(j-1)}| \leq \rho_i \quad (i=1,2,3,...,n) \quad (5.31)$$

$$|x_i^{(j)} - x_i^{(b)}| \leq \rho_i \quad (i=1,2,3,...,n) \quad (5.32)$$

burada,

τ ve ρ_i : hedef fonksiyonu ve dizayn değişkeni toleransları.

(5.29) ve (5.30) denklemleri hedef fonksiyonu değerleri arasındaki farkları, (5.31) ve (5.32) denklemleri ise dizayn değişkeni farklarını göstermektedir. Eğer yukarıdaki 4 denklem gerçekleşmezse, aşağıdaki iki duruma ulaşıldığında işlem durdurulacaktır.

$$n_s = N_s \quad (5.33)$$

$$n_{si} = N_{si} \quad (5.34)$$

burada,

n_s : Altproblem iterasyonlarının sayısı.

n_{si} : Art arda oluşan dizayn setlerinin sayısı.

N_s : Maksimum iterasyon sayısı.

N_{si} : Art arda oluşan, maksimum uygun olmayan dizayn setlerinin sayısı. [2]

5.2.1.4 Altproblem yaklaşımı metodu için bazı önemli hususlar

Hedef fonksiyonu ve SVs için yaklaşımlar kullanıldığı için, optimum dizayn bu yaklaşımlar ne kadar iyi ise o kadar iyi olabilir. İyi yaklaşım fonksiyonları elde etmek için bazı önemli hususlar aşağıda belirtmiştir:

- Altproblem yaklaşımında optimizier, hedef fonksiyonu ve SVs fonksiyonlarına yaklaşımlar sağlamak için rasgele dizaynlar oluşturur. Bu yüzden yaklaşım yavaş olabilir. Bazen birden fazla uygun dizayn seti ile başlayarak yaklaşımı hızlandırmak mümkündür. Bunu yapmak için basitçe, rasgele dizayn oluşturma aracını birkaç kez çalıştırıp, bütün uygun olmayan dizayn setlerini silmek yeterlidir. Buna alternatif bir yol olarak da, eğer problemin yapısı hakkında fikir sahibiyseniz, kabul edilebilir dizayn değişkenleri ile birkaç kez tek döngü çalıştırma aracı ile dizayn setleri elde edebilirsiniz.
- Eğer altproblem yaklaşımı metodu ile çok sayıda uygun olmayan dizayn oluşmuşsa, bu SV yaklaşım fonksiyonunun asıl fonksiyonu tam olarak temsil edemediği anlamına gelebilir. Bu durumda aşağıdakiler yapılabilir:
 - * Art arda oluşan maksimum uygun olmayan set sayısı artırılıp yeniden altproblem yaklaşımı çalıştırılır.
 - * Art arda yapılan altproblem yaklaşımı çalıştırmalarından elde edilen en iyi setleri periyodik olarak seçip daha iyi bir yaklaşım elde edilebilir.

5.2.2 Birinci mertebe (First Order) metodu

Bu metotta türevsel bilgiler kullanılarak hesaplamalar yapılmaktadır. Bölüm 5.2 de verilen formdaki formüller ile tanımlanan sınırlanmış problem durumu penaltı fonksiyonları yardımıyla sınırlanmamış probleme dönüştürülür. Hedef fonksiyonu ve SVs penaltı fonksiyonları için, arama yönünde kullanılmak üzere türevler oluşturulur. Yaklaşım sağlanana kadar her bir iterasyonda değişik "steepest descent" ve "conjugate direction" aramaları yapılır. Her bir iterasyon, arama yönü ve eğim hesaplamalarını içeren alt iterasyonlardan oluşur. Diğer bir deyişle, bir tane birinci mertebe dizayn optimizasyonu iterasyonu, birkaç analiz döngüsü (loop) yapacaktır. Altproblem yaklaşımı metoduna kıyasla, bu metot daha kesin sonuçlar verdiği için

daha fazla tercih edilebilir. Birinci merteye metodu OPTYPE,FIRST komutu ile seçilir. [2]

5.2.2.1 Sınırlanmamış hedef fonksiyonu

Bölüm 5.2 'de ki problemin sınırlanmamış hali aşağıdaki gibidir.

$$Q(x, q) = \frac{f}{f_0} + \sum_{i=1}^n P_x(x_i) + q \left(\sum_{i=1}^{m_1} P_g(g_i) + \sum_{i=1}^{m_2} P_h(h_i) + \sum_{i=1}^{m_3} P_w(w_i) \right) \quad (5.35)$$

burada,

Q : Sınırlanmamış hedef fonksiyonu (boyutsuz)

P_x, P_g, P_h , ve P_w : Sınırlanmış dizayn değişkenleri ve SVs için uygulanan penaltılar.

f_0 : Aktif dizayn setleri grubundan seçilen hedef fonksiyonu referans değeri.

Sınırlamaların sağlanması bir cevap yüzeyi parametresi olan q ile kontrol edilir.

Dış penaltı fonksiyonları (P_x), dizayn değişkenlerine uygulanır. SVs sınırlamaları genişletilmiş iç penaltı fonksiyonları (P_g, P_h, P_w) ile temsil edilir. Örneğin, üst limite sınırlanmış (5.8) denklemi için penaltı fonksiyonu şöyle yazılır:

$$P_g(g_i) = \left(\frac{g_i}{g_i + a_i} \right)^{2\lambda} \quad (5.36)$$

burada,

λ : Çok büyük bir tam sayıdır. Sınırlama ihlal edildiğinde fonksiyonun değeri çok büyük, sınırlama sağlandığında ise fonksiyonun değeri çok küçük olacaktır.

Kalan sınırlamalar için kullanılan penaltılar da benzer formdadır.

Arama yönü bulunduğunda, eğer Q fonksiyonu iki fonksiyonun toplamı şeklinde yeniden yazılırsa hesap bakımından kolaylık elde edilir. [2]

$$Q_f(x) = \frac{f}{f_0} \quad (5.37)$$

ve

$$Q_p(x, q) = \sum_{i=1}^n P_x(x_i) + q \left(\sum_{i=1}^{m_1} P_g(g_i) + \sum_{i=1}^{m_2} P_h(h_i) + \sum_{i=1}^{m_3} P_w(w_i) \right) \quad (5.38)$$

olmak üzere, daha sonra denklem (5.35) aşağıdaki formu alır.

$$Q(x, q) = Q_f(x) + Q_p(x, q) \quad (5.39)$$

Q_f ve Q_p fonksiyonları sırası ile hedef fonksiyonu ve penaltı sınırlamaları ile ilgilidir.

5.2.2.2 Arama yönü

Her bir optimizasyon iterasyonu (j) için bir arama yönü vektörü ($d^{(j)}$) bulunur. Bir sonraki iterasyon (j+1) aşağıdaki denklemle bulunur.

$$x^{(j+1)} = x^{(j)} + s_j d^{(j)} \quad (5.40)$$

Çizgi arama parametresi olan s_j , Q fonksiyonunun $d^{(j)}$ yönündeki minimum değerine karşılık gelir. s_j nin çözümü için, "altın bölge" (golden section) algoritması ile "yerel ikinci derece uygunluğu" (local quadratic fitting) tekniğinin bir kombinasyonu kullanılır. s_j nin sınırları şöyledir:

$$0 \leq s_j \leq \frac{S_{\max}}{100} s_j^* \quad (5.41)$$

burada,

s_j^* : Aktif iterasyonun çizgi araması için mümkün olan en büyük adım büyüklüğü (içeriden hesaplanır).

$S_{\max}$: Yüzde olarak, maksimum çizgi araması adım büyüklüğü.

Denklem (5.39) un genel minimumunun bulunmasında, arama yönünün ardışık olarak oluşturulması ve q parametresinin içeriden ayarlanması önemlidir. Başlangıç iterasyonu için (j=0) arama yönü, sınırlanmamış hedef fonksiyonunun eğiminin negatifi olarak kabul edilir.

$$d^{(0)} = -\nabla Q(x^{(0)}, q) = d_f^{(0)} + d_p^{(0)} \quad (q=1) \quad (5.42)$$

ve

$$d_f^{(0)} = \nabla Q_f(x^{(0)}) \quad \text{ve} \quad d_p^{(0)} = \nabla Q_p(x^{(0)}) \quad (5.43)$$

İlk iterasyon için arama metodu "steepest descent" dir. Takip eden iterasyonlarda (j>0), Polak-Ribiere tekrarlama formülüne göre "conjugate directions" oluşturulur.

$$d^{(j)} = -\nabla Q(x^{(j)}, q_k) + r_{j-1} d^{(j-1)} \quad (5.44)$$

$$r_{j-1} = \frac{[\nabla Q(x^{(j)}, q) - \nabla Q(x^{(j-1)}, q)]^T \nabla Q(x^{(j)}, q)}{|\nabla Q(x^{(j-1)}, q)|^2} \quad (5.45)$$

Tüm dizayn değişkeni sınırlamaları sağlandığında, $P_x(x_i) = 0$ dır. Bu yüzden q parametresi Q_p fonksiyonundan çıkarılıp şöyle yazılabilir:

$$Q_p(x^{(j)}, q) = q \cdot Q_p(x^{(j)}) \quad \text{eğer} \quad \underline{x}_i \leq x_i \leq \bar{x}_i \quad (i=1,2,3,...,n) \quad (5.46)$$

Eğer uygun düzeltmeler yapılırsa, (5.44) denkleminin conjugate yapısı bozulmaksızın q iterasyondan iterasyona değiştirilebilir. q parametresini ayarlamak, yaklaşım sağlandığında SVs sınırlamalarının iç kontrolünün (gerektiğinde limit değerlerine itmek) yapılmasını temin eder. Denklem (5.44) iki yön vektörüne ayrıldığında bu ayarlama daha net olarak görülebilir:

$$d^{(j)} = d_f^{(j)} + d_p^{(j)} \quad (5.47)$$

burada her bir yön ayrı bir tekraralama ilişkisine sahiptir:

$$d_f^{(j)} = -\nabla Q_f(x^{(j)}) + r_{j-1} d_f^{(j-1)} \quad (5.48)$$

$$d_p^{(j)} = -q \nabla Q_p(x^{(j)}) + r_{j-1} d_p^{(j-1)} \quad (5.49)$$

Algoritma bazen $r_{j-1}=0$ yapılarak yeniden başlatılır. Yeniden başlatma işlemi, kötü durumlarda, yaklaşımın sağlanmasına az kaldığında veya kritik sınırlamaların sağlanmasında hassas olduğunda yapılır.

Eğim vektörü aşağıdaki gibi bir yaklaşımla bulunabilir:

$$\frac{\partial Q(x^{(j)})}{\partial x_i} \approx \frac{Q(x^{(j)} + \Delta x_i e) - Q(x^{(j)})}{\Delta x_i} \quad (5.50)$$

burada,

e : Sadece i . terimi için 1, diğer bütün terimleri için 0 olan vektör.

$$\Delta x_i = \frac{\Delta D}{100} (\bar{x}_i - \underline{x}_i)$$

ΔD : ileri fark adım büyüklüğü. [2]

5.2.2.3 Yaklaşım

Birinci mertebe iterasyonları yaklaşım sağlanana veya işlem kesilene kadar devam eder. Bu iki olay her iterasyonun sonunda kontrol edilir.

Aktif iterasyon dizayn setinin (j), bir önceki setle ($j-1$) ve en iyi dizayn seti ile (b) aşağıdaki kıstaslara göre karşılaştırılmasıyla yaklaşımın sağlandığı kabul edilir.

$$|f^{(j)} - f^{(j-1)}| \leq \tau \quad (5.51)$$

$$|f^{(j)} - f^{(b)}| \leq \tau \quad (5.52)$$

burada,

τ : Hedef fonksiyonu toleransı

İşlem aşağıdaki durum gerçekleştiğinde kesilir.

$$n_i = N_1 \quad (5.53)$$

burada,

n_i : İterasyonların sayısı.

N_1 : Müsaade edilen maksimum iterasyon sayısı. [2]

5.2.2.4 Birinci mertebe metodu için bazı önemli hususlar

Altproblem yaklaşımı metoduna kıyasla birinci mertebe metodunun doğruluk derecesi daha yüksektir. Fakat yüksek hassasiyet en iyi çözümü vermeyi garanti etmez. Bununla ilgili dikkate alınması gereken bazı önemli hususlar şunlardır:

- Uygun olmayan bir dizayna yaklaşım sağlaması mümkündür. Bu durumda, bir yerel minimum bulmuş olabilir veya uygun dizayn uzayı bulunmayabilir. Eğer böyle bir durum oluşursa, dizayn uzayı hakkında daha iyi bir ölçü olan altproblem yaklaşımı metodu kullanmak daha faydalı olabilir. Başka bir yol olarak da, rasgele dizaynlar oluşturup (eğer varsa), bunlardan birisini kullanarak birinci mertebe metodu yeniden çalıştırılabilir.
- Birinci mertebe metodunun bir yerel minimuma ulaşması ihtimali yüksektir. Eğer başlangıç noktası bir yerel minimuma çok yakınsa, genel minimum yerine bu noktayı bulabilir. Eğer bir yerel minimum bulunduğundan şüphe ediliyorsa yukarıda anlatıldığı gibi rasgele dizaynlar oluşturulabilir veya altproblem yaklaşımı metodu çalıştırılabilir.

- Çok dar olan bir hedef fonksiyonu toleransı, çok sayıda iterasyon oluşmasına neden olacaktır.

5.3 Optimizasyon Araçları

5.3.1 Tek döngü analiz aracı (Single loop analysis tool)

Bu, dizayn uzayını anlayabilmek için kullanılan çok basit bir araçtır. Gerekli olmasa da SVs ve hedef fonksiyonu değerlerini hesaplamak faydalı olacaktır. Dizayn değişkenlerinin tamamı kullanıcı tarafından tanımlanır. Bir tek döngü, bir adet bütün sonlu eleman analizine eşittir. OPTYPE,RUN komutu ile belirtilebilir.

Her bir iterasyonun başlangıcında kullanıcı dizayn değişkenlerini değerlerini tanımlar.

$x = x^*$ = Kullanıcının tanımladığı dizayn değişkeni değerleri

Bu değişken değerleri ile bir adet döngü yapılır. Eğer SVs veya hedef fonksiyonu tanımlanmışsa g_i^* , h_i^* , w_i^* ve f^* değerleri hesaplanır. [2]

5.3.2 Rasgele dizayn aracı (Random tool)

Bu dizayn aracı her bir iterasyonda dizayn değişkeni vektörünü rasgele değerlerle doldurur ve eğer tanımlanmışsa o andaki vektöre göre SVs ve hedef fonksiyonu değerlerini hesaplar. OPTYPE,RAND komutu ile seçilir ve aşağıdaki durumlar oluşana kadar çalışır.

$$n_r = N_r \quad (5.54)$$

$$n_r = N_f \quad \text{eğer } N_f \geq 1 \text{ ise} \quad (5.55)$$

burada,

n_r : Her bir çalıştırmada oluşturulan rasgele iterasyonları sayısı

n_f : Toplam uygun dizayn sayısı (önceki uygun setleri de içerir).

N_f : Belirtilen maksimum iterasyon sayısı

N_f : Arzulanan uygun dizayn seti sayısı [2]

5.3.3 Süpürme aracı (sweep tool)

Süpürme aracı kullanıcı tarafından tanımlanan bir referans dizayn seti merkezinde genel dizayn uzayını taramak için kullanılır. OPTYPE,SWEEP komutu ile seçilir.

Araç çalıştırıldığında, diğer bütün dizayn değişkenlerini sabit tutarak, her bir dizayn değişkeni için bir süpürme veya diğer bir deyişle tarama yapar. Her bir süpürme işleminde SVs ve hedef fonksiyonu hesaplanıp saklanır.

Araç bir kere çalıştırıldığında n_s adet dizayn seti hesaplar.

$$n_s = n.N_s \quad (5.56)$$

burada,

n : Dizayn değişkenlerini sayısı.

N_s : Kullanıcı tarafından belirtilen ve her bir dizayn değişkeni yönünde yapılacak işlemler. [2]

5.3.4 Faktöriyel aracı (Factorial tool)

Bu istatistiksel araç bütün uç noktadaki durumları görmek için kullanılır.

OPTYPE,FACT komutu ile seçilir. Daha sonra OPFACT komutu ile, dizayn uzayının bütün ve kesirli olmak üzere iki çeşit faktöriyel değerlendirmesi yapılır. n adet dizayn değişkeninin bir bütün faktöriyel değerlendirmesi n_f adet dizayn seti oluşturur.

$$n_f = 2^n \quad (5.57)$$

dizayn değişkeni vektörünün her bir elemanı iki uç nokta değeri olacaktır. Bunlar:

$$x_i = \bar{x}_i \text{ veya } \underline{x}_i \text{ dir.} \quad (5.58)$$

Bu yüzden bir bütün faktöriyel değerlendirmede, dizayn değişkeni uç değerlerinin her eşleşmesi n boyutlu bir dizayn uzayında dikkate alınır.

Bir kesirli faktöriyel değerlendirmede oluşan dizayn setlerinin sayısı aşağıdaki gibidir:

$$n_f = 2^n / M \quad (M = 2, 4, 8, \dots) \quad (5.59)$$

Bu nedenle, bir 1 / 2 kesirli faktöriyel değerlendirme (M=2), bütün faktöriyel değerlendirmenin dizayn seti sayısının yarısıdır. [2]

5.3.5 Eğim aracı (Gradient tool)

Bu araç SVs ve hedef fonksiyonunun, dizayn değişkenlerine göre eğimlerini hesaplar. OPTYPE, GRAD komutu ile seçilir. Eğimin oluşturulacağı nokta için bir referans dizayn seti tanımlanır. Bu set OPGRAD komutunun DSET argümanı ile belirtilir. Hedef fonksiyonu için referans durum aşağıdaki gibi olsun;

$$f_r(x) = f(x^{(r)}) \quad (5.60)$$

Hedef fonksiyonunun eğim şöyledir;

$$\nabla f_r = \left[\frac{\partial f_r}{\partial x_1}, \frac{\partial f_r}{\partial x_2}, \dots, \frac{\partial f_r}{\partial x_n} \right] \quad (5.61)$$

Her bir dizayn değişkenine göre, eğim için aşağıdaki gibi yaklaşım sağlanabilir.

$$\frac{\partial f_r}{\partial x_i} = \frac{f_r(x + \Delta x_i e) - f_r(x)}{\Delta x_i} \quad (5.62)$$

burada,

e : i. Eleman için 1, diğer bütün elemanlar için 0 olan vektör.

$$\Delta x_i = \frac{\Delta D}{100} (\bar{x}_i - \underline{x}_i)$$

ΔD : İleri fark adım büyüklüğü (yüzde olarak)

Benzer hesaplamalar bütün SVs için yapılır. [2]



BÖLÜM 6

SONUÇLAR ve ÖNERİLER

Bir dizayn optimizasyonu problemi için en önemli basamak şüphesiz ki problemi doğru olarak formüle etmek, nelerin dizayn değişkenleri (DV's), nelerin dizayn sınırlamaları (SV's), ve neyin hedef fonksiyonu olarak ANSYS programına tanıtılacağına karar vermektir.

Eğer kullanıcı ANSYS komutlarına yeterince hakimse, analizin tüm safhalarını bir batch dosyası oluşturup bunu ANSYS'den çağırarak, değilse interaktif olarak (GUI: Graphical User Interface) problemi çözmesi daha uygun olacaktır.

Geometrik model oluşturulurken, özellikle dizayn değişkeni olarak tanımlanacak olan boyutların sabit sayılar atayarak değil, parametrik olarak tanımlanmasına dikkat edilmelidir.

Sonu elemanlar analizini yaptıktan sonra dizayn sınırlamalarını (SV's) ve hedef fonksiyonunu tanımlamak için eleman tabloları kullanılır. Özellikle bu tabloların oluşturulma yollarının üzerinde hassasiyetle durmak dizayn sınırlamalarını ve hedef fonksiyonlarını doğru olarak tanımlamak için çok faydalı olacaktır.

Dizayn değişkenlerinde olduğu gibi, eleman tablolarından elde edilen dizayn sınırlamaları ve hedef fonksiyonunun da parametrik olarak tanımlanmasına dikkat edilmelidir.

ANSYS'in dizayn optimizasyonu problemlerini çözmek için altproblem yaklaşımı metodu (subproblem approximation method) ve birinci mertebeli metodu (first order method) olmak üzere kullandığı iki temel metottan hangisinin kullanılacağına karar

verilmelidir. Bilinmelidir ki; altproblem yaklaşımı metodu belirli sayıda uygun dizayn setinden faydalanarak, hedef fonksiyonu ve sınırlama fonksiyonlarına yaklaşımlar oluşturur ve yaklaşım fonksiyonları gerçek fonksiyonlara ne kadar yaklaşıksa sonuçlar o kadar kesin olabilir. Bunun için optimizasyon araçları da kullanılarak çok sayıda uygun set elde edilirse yaklaşım daha iyi olacaktır. Birinci mertebe metodunda ise altproblem yaklaşımı metoduna göre daha hassas sonuçlar elde edilebilir.

ANSYS'in kullandığı metotların algoritmalarını anlamak problemleri daha iyi kavramak için faydalı olacaktır. Her iki metotta da sınırlanmış haldeki problem, sınırlanmamış problemlere dönüştürülür ve elde edilen fonksiyon değişik arama metotları kullanılarak minimize edilir. Bunun için hedef fonksiyonuna sınırlamalardan kaynaklanan penaltı fonksiyonları eklenir. Arama metotları ve penaltı fonksiyonları hakkında geniş bilgi edinmek için ARORA, (1989) ve HAFTKA, GÜRDAL, KAMAT, (1990) gibi kaynak kitaplardan faydalanılabilir. Bunların yanında dizayn optimizasyonunun teorisi hakkında yazılmış başka kitapları da incelemek faydalı olacaktır.

KAYNAKLAR

- [1] Arora, J., (1989), Introduction to Optimum Design, McGraw-Hill, Inc.
- [2] ANSYS Release 5.3 Help System, (1996), ANSYS Inc.
- [3] HAFTKA, R., GÜRDAL, Z., KAMAT, M.P., (1990), Elements of Structural Optimization, Kluwer Academic Publishers.
- [4] SINHA, D.P., HSU, T.R., (1992), Computer Aided Design, West Publishing Company
- [5] FYFE, K., McKAY, B., MURADALI, A., SORENSEN, C., GEISLER, K.J., OLOFSSON, T., SKOCZYLAS, P., TOOGOOD, R., ANSYS Tutorials, <http://www.mece.ualberta.ca/tutorials/ansys/tutorial.html> ,
<http://www.mece.ualberta.ca/tutorials/ansys/optimization/optimization.html> ,
University of Alberta Mechanical Engineering

Tablo A.1- ANSYS Optimizasyon modülünde kullanılan komutlar [2]

OPEQN	Altproblem yaklaşımı metodu için eğri uygunluğunu ayarlar
OPFACT	Faktöriyel değerlendirme aracının tipini tanımlar
OPFRST	Birinci mertebe metodunun ayarlarını tanımlar
OPGRAD	Gradyen aracı için hangi dizayn setinin kullanılacağını belirtir
OPKEEP	En iyi set sonuçlarının ve veritabanı dosyasının saklanıp saklanmayacağını belirtir.
OPLOOP	Optimizasyon döngü ayarlarını kontrol eder
OPPRINT	Detaylı optimizasyon çıktısını aktive eder
OPRAND	Rasgele aracı için maksimum iterasyon sayısını belirtir
OPSUBP	Altproblem yaklaşımı metodu için iterasyon sayısını belirtir
OPSWEEP	Süpürme aracı için referans noktası ve değerlendirilecek nokta sayısını belirtir.
/OPT	Optimizasyon modülüne girişi sağlar
OPTYPE	Kullanılacak optimizasyon metodunu belirtir
OPUSER	Kullanıcı kaynaklı dış optimizasyon metodunu tanımlar
OPVAR	Optimizasyon değişkenlerini belirtir
OPADD	İki seti ekleyerek başka bir set oluşturur
OPCLR	Optimizasyon veritabanını siler
OPDEL	Optimizasyon dizayn setlerini siler
OPMAKE	Skalar parametrelerin aktif değerlerini kullanarak bir dizayn seti oluşturur
OPSEL	Takip eden optimizasyon döngüsü için belirtilen dizayn setlerini seçer
OPANL	Optimizasyon döngülerinde kullanılacak analiz dosyasını belirtir
OPDATA	Optimizasyon verilerinin saklanacağı dosyayı belirtir
OPRESU	Optimizasyon veritabanına belirtilen dosyadan veri okur
OPSAVE	Bütün optimizasyon verilerini bir dosyaya yazar
OPEXE	Optimizasyon döngülerini başlatır
OPLFA	Faktöriyel aracının sonuçlarını gösterir
OPLGR	Gradyen aracının sonuçlarını grafik olarak çizer
OPLIST	Dizayn setlerini listeler
OPLSW	Genel süpürme aracının sonuçlarını grafik olarak çizer
OPRFA	Faktöriyel aracının sonuçlarını yazar
OPRGR	Gradyen aracının sonuçlarını yazar
OPRSW	Genel süpürme aracının sonuçlarını yazar
PLVAROPT	On taneye kadar parametreyi grafik olarak çizer
PRVAROPT	On taneye kadar parametreyi listeler
XVAROPT	X eksenini olarak çizilecek parametreyi belirtir

ÖZGEÇMİŞ

1976 İstanbul doğumlu Engin AKHOROZ, 1992 yılında Şişli Endüstri Meslek Lisesi, Elektrik Bölümünü bitirmiştir. Aynı yıl girdiği İstanbul Teknik Üniversitesi, Sakarya Mühendislik Fakültesi, Makina Mühendisliği bölümünden 1996 yılında mezun olmuştur. 1996 yılında girdiği İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Makina Mühendisliği Anabilim Dalı, Konstrüksiyon Programından, ilk yılı yine aynı üniversitenin Dil ve İnkılap Tarihi Bölümü bünyesinde İngilizce hazırlık eğitimi olmak üzere 1999 yılında mezun olmuştur.



T.C. YÜKSEKÖĞRETİM KURULU
Yükseköğretim Kurulu Başkanlığı