

**T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

İnşaat Mühendisliği Anabilim Dalı

İnşaat Mühendisliği Programı



**ÇELİK İLE GÜÇLENDİRİLMİŞ BETONARME YAPILARIN ANALİZ
DEĞERLENDİRİLMESİ**

YÜKSEK LİSANS TEZİ

Ekrem TÜFEKÇİ

İnşaat Mühendisliği Anabilim Dalı

İnşaat Mühendisliği Programı

Eylül , 2019

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



ÇELİK İLE GÜÇLENDİRİLMİŞ BETONARME YAPILARIN ANALİZ
DEĞERLENDİRİLMESİ

YÜKSEK LİSANS TEZİ

Ekrem TÜFEKÇİ

Y1713.090030

İnşaat Mühendisliği Anabilim Dalı

İnşaat Mühendisliği Programı

Dr. Öğr. Üyesi Sepanta Naimi

Eylül , 2019

ONAY FORMU



T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜ

Yüksek Lisans Tez Onay Belgesi

Enstitümüz İnşaat Mühendisliği Ana Bilim Dalı İnşaat Mühendisliği Tezli Yüksek Lisans Programı Y1713.090030 numaralı öğrencisi Ekrem TÜFEKÇİ'nin "ÇELİK İLE GÜÇLENDİRİLMİŞ BETONARME YAPILARIN ANALİZ DEĞERLENDİRİLMESİ" adlı tez çalışması Enstitümüz Yönetim Kurulunun 02.08.2019 tarih ve 2019/17 sayılı kararıyla oluşturulan jüri tarafından *ay.birliği* ile Tezli Yüksek Lisans tezi olarak *Kabul* edilmiştir.

Öğretim Üyesi Adı Soyadı

İmzası

Tez Savunma Tarihi : 13/09/2019

1) Tez Danışmanı: Dr. Öğr. Üyesi Sepanta NAİMİ

2) Jüri Üyesi : Doç. Dr. Orhan CANPOLAT

3) Jüri Üyesi : Dr. Öğr. Üyesi Kaveh DEGHANIAN

Not: Öğrencinin Tez savunmasında **Başarılı** olması halinde bu form **imzalanacaktır**. Aksi halde geçersizdir.

YEMİN METNİ

Yüksek Lisans Tezi Olarak Sunduğum “ Çelik İle Güçlendirilmiş Betonarme Yapıların Analiz Değerlendirilmesi ” adlı çalışmamda, tezimin proje kısmından sonuçlandığı zamana kadar tüm süreçlerde bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksızın yazıldığını ve yararlandığım eserlerin Bibliyografya’da gösterildiği gibi olduğunu, bunlara atıf yaparak yararlanılmış olduğunu belirtir ve beyan ederim. (.../.../2019)

Ekrem TÜFEKÇİ

ÖNSÖZ

İnşaat sektörü itibariyle insanların genellikle yapılar mimari açıdan bakıp görünüşüne önem verdiği ama arka planda strateji, kullanım ömrü, maliyet, ekonomi, plan ve program açısından bakıldığında yapı birçok koşul ve faktöre bağlı olarak etkilenip değişkenlikle insanları birçok faktörde etkileyen bir sektördür. İnsanların olduğu her alanda inşa edilmekte olan birçok yapı vardır ve bu yapıların kullanılabilirliğinin herbiri insanları etkilemektedir. Bu yüzden de coğrafi konum sebebiyle fay hatları üzerinde bulunan her yapı, kullanım ömrü ve süreci boyunca uygun bakım ve onarımlara ihtiyaç duymaktadır. Bu süreç ve ekonomisi esasına bakıldığında yapının kullanım süreci esnasında yapının ekonomi açısından insanları fazla refaha erdirmesi için çelik ile güçlendirilmiş betonarme yapıların tasarımı ve kullanımı esasına bakıldığında mühendislik açısından önemi göz önünde bulundurularak bu çalışma yapılmıştır. Yüksek lisans tezi olarak sunduğum bu çalışmada "ÇELİK İLE GÜÇLENDİRİLMİŞ BETONARME YAPILARIN ANALİZ DEĞERLENDİRİLMESİ" adlı çalışmamda bana yardımcı olan hocam Dr. Öğr. Sepanta NAİMİ'ne teşekkür ederim. Ayrıca, en büyük teşvik ve sabırla hayatım boyunca bana güç veren sevdiklerim babama, anneme, kardeşime ve sevdiğime sunuyorum.

Eylül, 2019

Ekrem TÜFEKÇİ

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xvii
ABSTRACT	xix
1. GİRİŞ	1
2. METODOLOJİ	5
2.1 Tasarım Gereksinimler Ve Süreçleri	8
2.2 Malzeme Seçimi	11
2.3 Yapının Davranışı	13
2.4 Yapıya Ait Yükler	16
2.5 Deprem	17
3. ANALİZ VE SONUÇLAR	21
3.1 AutoCad Tasarım Görüntüsü	21
3.2 Seismo Struct Programı Tasarım Görüntüleri	22
3.3 Seismo Struct Programı Analiz Sonuçları.....	24
3.4 Sap2000 Programı Tasarım Görüntüleri	27
3.5 Sap2000 Programı Analiz Sonuçları	27
3.6 ProBina Tasarım Görüntüleri	29
3.7 ProBina Yapı analizi	33
3.7.1 ProBina Döşeme Yük Analizi.....	34
3.7.2 Döşeme Yükleme Grafikleri	36
4. SONUÇ VE ÖNERİLER.....	41
5. KAYNAKÇA	45
EKLER.....	49
ÖZGEÇMİŞ.....	239

KISALTMALAR

TBDY	: Türkiye Bina Deprem Yönetmeliđi
DBYBHY	: Deprem Bölgesinde Yapılacak Binalar Hakkında Yönetmeliđi
ÇİGBY	: Çelik ile Güçlendirilmiş Betonarme Yapılar
AFAD	: Afet ve Acil Durum Yönetim Başkanlığı
SLR	: Çelik Lamine Kauçuk Mesnet
RB	: Kauçuk Mesnet
HDNR	: Yüksek Sönümlü Doğal Kauçuk Mesnet
PGA	: En Büyük Yer İvmesi
PGV	: En Büyük Yer Hızı
Ss	: Harita Spektral İvme Katsayısı
S1	: 1.0sn periyod için Harita Spektral İvme Katsayısı

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 2.1: Yapım Aşamaları.....	9
Çizelge 2.2: Tasarım Spektrum parametreleri	18
Çizelge 3.1: Sap 2000 Analiz Programı	28

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: Çelik Çekme Grafiği	12
Şekil 2.2: Çelik Çekme Grafiği-2	13
Şekil 2.3: Yapının Davranışı.....	14
Şekil 2.4: Yapının Davranışı.....	14
Şekil 2.5: Yapının Davranışı.....	15
Şekil 2.6: Yapının Davranışı.....	15
Şekil 2.7: Afad Deprem Fay Hattı (Afad, 2018)	19
Şekil 3.1: Autocad programından 12 katlı tasarımın yapının ön görünüşü	21
Şekil 3.2: Seismo Struct programından 6 katlı tasarımın yapının ön görünüşü	22
Şekil 3.3: Seismo Struct programından 6 katlı tasarımın yapının çelik ve betonarme bilgilerinin ön görünüşü	22
Şekil 3.4: Seismo Struct programından 6 katlı tasarımın yapının donatı kesiti	23
Şekil 3.5: Seismo Struct programından 6 katlı tasarımın yapının donatı kesiti	23
Şekil 3.6: Seismo Struct programından Kocaeli depremin Kandilli/İstanbul merkezli deprem hareket yüzeyi.....	24
Şekil 3.7: Sap2000 programından 6 katlı tasarımın yapının ön görünüşü.....	27
Şekil 3.8: ProBina Temel kodu deplasman dayanımı.....	29
Şekil 3.9: ProBina Temel kodu moment-x yönü dayanımı	30
Şekil 3.10: ProBina Temel kodu moment-y yönü dayanımı	30
Şekil 3.11: ProBina Temel kodu moment-x ve moment-y yönü dayanımı	31
Şekil 3.12: ProBina 1.Kat kodu deplasman dayanımı	32
Şekil 3.13: ProBina 1.Kat kodu moment-x yönü dayanımı.....	32
Şekil 3.14: ProBina 1.Kat kodu moment-y yönü dayanımı.....	33
Şekil 3.15: ProBina 1.Kat kodu moment-x ve moment-y yönü dayanımı.....	33
Şekil 3.16: ProBina Yapı kodlarının moment x-y göstergeleri	34
Şekil 3.17: ProBina Yüklemeler (Kombinasyon G+QF).....	36
Şekil 3.18: ProBina Yüklemeler (Kombinasyon G+Sx+)	36
Şekil 3.19: ProBina Yüklemeler (Kombinasyon G-Sx-)	37
Şekil 3.20: ProBina Yüklemeler (Kombinasyon G-Sy+)	38
Şekil 3.21: ProBina Yüklemeler (Kombinasyon G-Sy-)	39

ÇELİKLE İLE GÜÇLENDİRİLMİŞ BETONARME YAPILARIN ANALİZ DEĞERLENDİRİLMESİ

ÖZET

Günümüzde yapıların genellikle sadece betonarme üzerine yapılması kullanım ömrü, güvenliği ve çevreye verdiği zarar göz önünde bulundurulmadan yapılan yapılarla beraberinde çoğunlukla kullanım ömrü doldurulmadan ya restorasyon ya da yıkılarak yeniden yapım aşamalarına gelmektedir. Restorasyon aşaması ve yapımının yaklaşık maliyeti yeniden yapım maliyetine yakın olmasından dolayı da çoğunlukla yıkıp yeniden yapılmaktadır. Bu da hem hafriyat hem de ekstra işçilik ve malzeme olduğundan dolayı bu da çevreye verilmiş ekstra zarar vermektedir. Bunlar ile alakalı elde edilen verileri sonuç bölümünde elde edebilirsiniz. Kontrolleşen yapılaşma, insan ihtiyacını karşılayacak herhangi bir yer veya konumda yapılaşması sonucunda zamanla çevre ihtiyaçları sebebiyle yapılaşma kontrolsüz yapılmaktadır. Bu yüzden yapılmış olan ya da yenileme geçirmiş yapılarımızın kullanıma açılması veya yapı inşa esnasında kullanılabilirliğini yada değiştirilebilirliğini artırabilecek güvenli yapılar inşa edilmesi gerekmektedir. Ya da yapının tasarım ömrü boyunca ihtiyaçları karşılaması için doğaya duyarlı yapı seviyesine getirilmek istendiğinde güçlendirme ya da yenileme ihtiyacı olduğunda maliyetleri sürekli artış göstermektedir. Eğer yapılarını geleceğe dönük tasarımı, kullanımını ve planlanması yapının kullanılabilirliğini göstermektedir. Biz yapılarımızı insanların kullanımına açtıktan sonra faal dışı duruma çevirmemek için kullanılabilirliğini ve kullanılabilirliğini üst seviyede tutulması gerekmektedir. Mühendislik için gerekli ve uygun olan yapı; tasarım ve kullanım esnasında güvenilir olması, ekonomik olması, geleceğe dönük olması, kullanılabilir olması ve diğer durumları içermesi gerekmektedir. İnsanların yaşam ve konfor standartlarını yüksekte tutacak ve yukarıda da bahsedildiği gibi yapının her koşulu sağlaması gerekmekte ve bununda ekonomik olması gerekmektedir. Yapının ilk aşaması olan tasarım ve projelendirme esnasında ekonomik anlamında planlama da yapılmaktadır. Bu planlama yapının tasarım ömrü boyunca olduğundan ve yapı kullanım süresi boyunca yeniden maliyetler oluşturmaması için güvenlikten ödün verilmeyecektir. TBDY 2018 yönetmeliğine uygun yük ve koşullar altında değerlendirilirken deprem ve yangın gibi doğal afetler de değerlendirilerek Türkiye de Kocaeli depremi baz alınarak yapılan tasarımların analizlerinin değerlendirilmesi sonucunda elde edilen verileri içermektedir. Yapı 3 çekirdeği de betonarme olduğunda yapının tasarım ömrü boyunca yorulmalarda düşünüldüğünde güçlendirmeye ya da süneklikten taviz vererek daha rijit yapı inşa edilmesi gerekmektedir. Bu da Türkiye yönetmelik gereğince mümkün olmamakla beraber yapı deprem gibi doğal afet koşulları altında zayıf düşmeden çözebilmek için ÇİGB yapı daha uygulanabilmektedir.

Anahtar Kelime: *Güçlendirilmiş Betonarme, Çelikle Güçlendirilmiş Betonarme, Çelik, Kompleks Yapı.*

ANALYSIS EVALUATION OF FIRST STRENGTHENED REINFORCED CONCRETE STRUCTURES WITH STEEL

ABSTRACT

Nowadays, structures are generally built only on reinforced concrete, and with the structures made without considering the service life, safety and harm to the environment, its often come to the stages of restoration or demolition without being without filling service life. However since the approximate cost of the restoration phase and construction is close to the reconstruction cost, it is often demolished and rebuilt. This is both earthmoving and extra workmanship and material, which causes extra damage to the environment. As a result of the structured construction in any place or location that will meet the human needs, the structured construction is uncontrolled due to environmental needs. Therefore, the structures that have been constructed or have been renovated have to be opened to use, or secure structures that can increase their usability or interchangeability during the construction of the building are required. Or, when it is desired to bring the structure to the nature-sensitive structure level to meet the needs throughout the design life, the costs increase continuously when there is a need for reinforcement or renovation. The future-oriented design, use and planning of your structures show you the usefulness of the structure. We need to keep our usefulness and availability at the highest level in order not to turn our structures into inactive state after making them available to people. The necessary and appropriate structure for engineering; must be reliable, economical, future-proof, usable and include other situations during design and use. People will maintain high standards of living and comfort, and as mentioned above, the building must meet every condition and be economical. During the first stage of the structure, design and project planning is also made in economic terms. As this planning is throughout the design life of the building and security will not be compromised so that it does not re-create costs during the life of the building. The Turkey Building of Earthquake Regulation comprises natural disasters given the assessment data obtained as a result of the evaluation of Turkey be based on the analysis of the designs Kocaeli earthquake as earthquake and fire regulations are invested in 2018 under a suitable load and conditions. When the structure of the structure 3 is reinforced concrete, it is necessary to construct a more rigid structure by making concessions from strengthening or ductility when considering the fatigue during the design life of the structure. This structure under conditions such as natural disasters, earthquakes, although not able to Turkey in accordance with the regulations can be applied to solve CIGB more structure than its weakening.

Keyword: *Reinforced Concrete, Steel Reinforced Concrete, Steel, Complex Structure.*

1. GİRİŞ

Beton yapıların kusurlarından bir tanesi de kesme veya kırılma dayanımlarının az olmasından dolayı güçlendirilmeye ihtiyaç duyulmaktadır. Betonarme yapıların deformasyon karşı koruyabilmek için TBDY (Türkiye Bina Deprem Yönetmeliği, 2018) uygun olacak şekilde tasarım ve imalat önemli rol almaktadır. TBDY belirlenmiş sınırlar içerisinde inşa esnasında uygulanabilirliğine göre inşaat mühendisi güçlendirme yada değişiklikler yapabilmektedir. Bu akademik çalışma içerisinde çelik ve iç çelik olarak adlandırabileceğimiz çelik ile güçlendirilmiş betonarme yapıların analiz değerlendirmesi yapılacaktır. Uygulanan TBDY yönetmelik gereğince iç çelik olarak adlandıracağımız betonarme içerisinde kalan çelik malzemesinin teoride olan uygulanabilirliğinin pratikte mümkün değildir. Bu yüzden de bazı durum ve koşullarda yapı inşaat mühendisi kontrolünde değişiklikler yapılabilmektedir. Bu yapıyı pratikte sorun olmadığı gibi gözüksede yönetmeliğe uygun değildir. Ama bu çelik ile güçlendirilmiş betonarme yapılar için geçerli değildir. Çelik malzemesi yüksekliği, kalınlığı, şekli, et kalınlığı, rijitliği, dayanımı vb herşey hazır tasarlanarak planlandığından bu pratikte karşılaşılan sorunlarla denk gelmemiş olunur(D. Michael, 2018). Ayrıca dinamik yüklere karşı betonarme yapı ile çigb yapı dayanımlı olduğu analizler sonucunda elde edilmiştir. Bu akademik çalışma içerisinde 2 farklı yapının analizlerinin değerlendirilmesi yapılmıştır. Her 2 yapıda 6 katlı ve 3 çekirdekten oluşmaktadır. 1.yapı tipimiz sadece betonarme olup 2.yapı ise orta çekirdek betonarme diğer sağ ve sol çekirdekler çelik ile inşa edildiğinde mühendislik açısından etkileri incelenmiştir. Yapılarda diğer bütün herşey aynı tasarlanmıştır. Yapılarda sonlu elemanlar tekniği ve cross yöntemi kullanılarak hesaplamalar yapılmıştır. Çalışma içeriğinde yüksek dayanıma sahip rijtlik sınırı aşmadan betonarme olup 1.yapı tamamı betonarme olduğundan dilatasyon boşluğu, bitişik nizam boşluğu ve dinamik hareketliliği etkilememesi için önlemler alınarak tasarlanmıştır. 2.yapı da tasarımında ise temel radye temel üzerine sağ ve sol çekirdekler çelik

yapı olarak tasarlanmıştır. Türkiye coğrafi konumu gereği bulunduğu fay hattı üzerinde yakın zamanda olmuş olan Kocaeli depremi baz alınmıştır. TBDY ve DBYBHY yönetmeliklerine göre deprem kuvveti artırılır ve etki eden kuvvetler küçültülür ve ona göre tasarım daha rijit halde tasarlanması planlanır. TBDY göre bakıldığında DD-1 (2475 yıl), DD-2 (475 yıl), DD-3 (72 yıl) ve DD-4 (50 yıl) (İstanbul Teknik Üniversitesi Deprem Mühendisliği ve Afet Yönetimi Enstitüsü,2017) baz alınarak deprem hesaplaması yapılmıştır. Ayrıca bu çalışmada aynı yöntemle beraber deprem kuvveti yönetmeliğinde üstüne çıkarak %50 büyütülmüştür. Aynı şekilde yönetmeliğe uygun olarak düz S420 nervürlü demir donatı kullanırken TBDY uygun şekilde donatı sargısı ve bağlantı noktaları güçlendirilerek ek donatılar montajı yapılarak tasarlanmıştır. Tasarlanmış olan 1. ve 2. yapılara Kocaeli deprem sırasıyla % 50 ,100, 140, 200, 250 olmak üzere 5 farklı misli kuvvetler uygulanmıştır. Kısacası yapı rijitliği ve güvenilirliği üst limitlere kadar zorlanmış olup yapılar sınırlar içerisinde hareket etmiştir. Komşu nizam yapı yada dilatasyon da zarar görülmemiştir. Yapılardan 1.yapı olan bütün çerçeveler betonarme yapı ağır hasar alırken restorasyon yada yeniden yapıma muhtaç duyulurken ÇİGB yapı onarımı mümkün olan az hasarlı yapı değerleri içerisinde kalmaktadır. Bu verilere göre betonarme yapı ve meliyeti insan güvenirliliği ve sağlığı açısından ve ekonomik açıdan artırılması gerekmektedir. Bunu ya yapının taşıyıcı sistemlerine dışarıdan sargı metodu yada sismik izolatör gibi yeni teknolojik malzemeler ile deprem etkisi azaltılarak çözülebilir veya yapı tasarımı esnasında donatı miktarı artırılabilir ve inşası buna uygun şekilde yapılır. Buradan da anlaşılacağı gibi bazı ince detayların dayanım ve can sağlığı konusunda etkisi yüksek olmuştur. Ayrıca farklı güçlendirme yöntemleri ile karşılaştırmalar yapılmıştır. Ayrıca hasar görmüş betonarme yapılar taşıma limit durum (TLD) ulaşmış yapı malzemelerin çelik halatlarla yada çelik plakalarla güçlendirilerek yapının 0 elemanlarını da çalıştırarak hem ekonomik hemde yapıyı hafifleten bir sistem çerçevesi üzerinden ilerlenerek bazı seçimleri yapılmıştır. Betonarme yapıların uygulanabilirliği ve narin malzemeye sahip olduğundan imalattaki her koşuldan etkilenir ve kullanılabilirliği esnasında malzeme narinliği artar. Bu sebepten dolayı yapıların hasar görmesi ve kullanılabilirliğini yitirmesi ve ekstra ekonomik yükler teşkil ettiğinden dolayı yapı deprem gibi doğal afetler yada dış etkenlerden hasar görerek Mühendislik

dışında yapıya dönüşmesi yerine imalat sırasında güçlendirilmiş betonarme yapılar eksiklerini tamamlayarak sorunları çözülebilmektedir. Eğer yapıların kullanılabilirliğini ve ömrünü artırarak sorun çözümüne gidilirse maliyet söz konusu olduğunda can ve mal kaybını da göz önüne alındığında çelikle güçlendirilmiş betonarme yapılar daha verimli çalıştığı gözlemlenmiştir. Bu verim taşıyıcı yükleri TLD sınırına ulaştıktan sonra geri dönüşü olmayan yada maliyeti daha fazla olan çözümlerden kurtarabilmek için güçlendirilmiş çelik yapıların kullanımı TBDY göre uygun görülebilirken DBYBHY göre cevap alınmasada derin yada T kesit kiriş olarak adlandırılabilir olan betonarme yapıların dayanımı artırmak istendiğinde eğilme donatısının olduğu yere çelik malzemelerinden dayanım elde edilerek çözümlenebilmekteydi. Ama bu betonarme yapının sünekliliğini azalttığından yapı rijit olmasına sebep oluşturmuş olup yapının istenilen davranış tipinden farklı yöntemlere geçmesi mümkün görünmekteydi. Bu yüzden yapının dayanım sınırın elastik betonarme yapı yerine elastik çelikle güçlendirilmiş betonarme yapı daha dayanımlı ve rijitlik sınırına ulaşmamış yapı demektir.

2. METODOLOJİ

Çelik malzemesinin 18.yy. sonlarına ve 19.yy. ortalarından itibaren kullanımına başlanmış ve özellikle endüstri ve endüstriyel devrimlerle beraber kullanımı artmış ve inşa edilen yapılarda kullanılmaya başlamasıyla başlayan süreç bu zamana kadar çeliğin içindeki gelişimi ve mimari nokta ülkelerin geliştirmesiyle standartlar oluşmuştur. Bu standartlar gelişmiş ülkeler tarafından oluşturulmuştur. Maalesef ülkemizde üretimi ve standart olmadığından maliyeti de söz konusunda aldığımızda hak ettiği itibarı ve yerini bulamamış, ekonomik nedenlerin yanı sıra tecrübe ve bilgi eksikliği nedeniyle de çelik kavramı ülkemizde diğer ülkelerle eş zamanlı gelişmemiştir. Bu eksiklik deprem kuşağında olup etkilendikten sonra ihtiyaç duyulup standart ve gelişimine öncelik verilmiş güncellemeler yapılmıştır.

Günümüzde yapıların genellikle sadece betonarme üzerine yapılması kullanım ömrü, güvenliği ve çevreye verdiği zarar göz önünde bulundurulmadan yapılan yapılarla beraber çoğunlukla kullanım ömrü doldurulmadan ya restorasyon ya da yıkılarak yeniden yapım aşamalarına gelmektedir. Ve restorasyon aşaması ve yapımının yaklaşık maliyeti yeniden yapım maliyetine yakın olmasından dolayı da çoğunlukla yıkıp yeniden yapılmaktadır. Bu da demek oluyor ki kullanım ömrü ya da kullanım ihtiyacı geleceğe dönük tasarlanmamış yapılar sonucunda yapılarımız maliyeti yüksek seviyelerdedir. Ayrıca temel ve zemin özelliklerine göre bölgelerde bazı değişiklik sebebiyle kullanım ömrü doldurmadan yapıların kullanım dışına bırakılmak zorunda kalabilmektedir. Yapılarımızın kullanım dışına bırakılması, ihtiyacın başka yerler ya da başka yapılar üzerinden yeniden oluşması bizlere ekstra maliyet ve kontrolsüz yapılaşmaya başlamaktadır.

Kontrolsüz yapılaşma, insan ihtiyacını karşılayacak herhangi bir yer veya konumda yapılaşması sonucunda zamanla çevre ihtiyaçları sebebiyle yapılaşma kontrolsüz yapılmaktadır. Bu yüzden yapılmış olan ya da restorasyon geçirmiş yapılarımızın kullanıma açılması veya yapı inşa esnasında kullanılabilirliğini yada değiştirilebilirliğini artıracak güvenli yapılar inşa edilmesi

gerekmektedir. Eđer yapı tasarımı konut olan yapının bir kısmı ticari alana çevrilmek istendiğinde yapının bu koşulları sağlaması gerekmektedir. Ya da yapının tasarım ömrü boyunca ihtiyaçları karşılaması için doğaya duyarlı yapı seviyesine getirilmek istendiğinde güçlendirme ya da restorasyon ihtiyacı olduğunda maliyetleri sürekli artış göstermektedir. Eđer yapılarımızı geleceğe dönük tasarımı, kullanımı ve planlanması yapının kullanılabilirliğini göstermektedir. Biz yapılarımızı insanların kullanımına açtıktan sonra faal dışı duruma çevirmemek için kullanılabilirliğini ve kullanılabilirliğini üst seviyede tutulması gerekmektedir. Bununda sebebi insanlar her geçen gün ihtiyaçlarının artması ve bundan kaynaklı dönüştürülebilirliğe uygun yapılarımızın olmasına ihtiyaç duyulmaktadır. Mesela bir otelde küçük bir yüz havuzu olduğunu düşünün ve insanların yoğun kullanımında çocuklara uygun koşullar sağlanmadığında ya da havuzda suyun ısınmasını sağlayamadığınızda o bölümün kullanılabilirliğini tasarlayamadığınızı kabul edilmesi gerekmektedir. Yapımızın her zaman, koşul ve durumda mühendislik için gerekli her imkânı sağlayabilmesi gerekmektedir. Yapının güçlendirilmesi gerektiğinde maliyeti yeniden yapımının maliyetine yakın olmaması gerekmektedir. Eđer maliyetler arasındaki fark azsa ya da insanların gayri safi millî hasılası yüksek olan yerlerde yeniden yapım aşamasına geçilmesi demektir bu da yapının kullanımında olan kişilerin taşınması, başka yerin kullanım süresi, yaşamdan kaybedilen zaman ve diğer durumlar göz önüne alındığında yapının dönüşüme uygun olması yapının kullanıcıları tarafından daha az maddi zarar yada manevi kayıp görmesi demektir.

Mühendislik için gerekli ve uygun olan yapı; tasarım ve kullanım esnasında güvenilir olması, ekonomik olması, geleceğe dönük olması, kullanılabilir olması ve diğer durumları içermesi gerekmektedir. İnsanların yaşam ve konfor standartlarını yüksek tutacak ve yukarıda da bahsedildiği gibi yapının her koşulu sağlaması gerekmekte ve bununda ekonomik olması gerekmektedir. Yapının ilk aşaması olan tasarım ve projelendirme esnasında ekonomik anlamında planlama da yapılmaktadır. Bu planlama yapının tasarım ömrü boyunca olduğundan ve yapı kullanım süresi boyunca yeniden maliyetler oluşturmaması için güvenlikten ödün verilmeyecektir. Bununla alakalı bilgiler yönetmeliklerimizde nasıl uygulanması gerektiği ile ilgili ayrıntılı bilgi verilmektedir. Yapının ekonomi açısından uygulanabilirliğini de inşaat mühendisleri planlamasını ve tasarımını yapacağından malzeme seçimi konusunda da titiz ve

ekonomik olması gerekmektedir. Bu konuda tez çalışmamda ayrıntılı bilgiler verdim. Malzeme seçimi sebebiyle yapı tasarım ekonomisi ve inşa ekonomisi arasında farklar olmakla beraber tasarım ömrü boyunca maddi ve manevi zarara uğramamak için seçilmiş olan malzemelerin kalitesi çok önemlidir. Tez çalışmamda 3 çekirdek oluşan bir yapıyı 2 farklı program üzerinden tasarlamış olup analizlerini değerlendirdiğimde yapıyı tasarımını ekonomik açıdan kullanılabilirliğinden dolayı tercih edilmesi malzemelerin tasarım önceliği seçilebilmektedir. 3 çekirdekli yapı demek, yapının tek döşeme üzerine inşa edilmiş olup yapının 3 farklı girişi üzerinden planlanmıştır. Yapı aynı koşullar altında analizler değerlendirildiğinde betonarme ve çelik-betonarme yapının ekonomi açısından ve geleceğe dönük yapı olma özelliklerini barındırdığında yapının tasarım esasları çelik-betonarme yapı daha uygulanabilirliğini görülmektedir.

Tez çalışması içerisinde TBDY 2018 yönetmeliğine uygun yük ve koşullar altında değerlendirilirken deprem ve yangın gibi doğal afetler de değerlendirilerek Türkiye de Kocaeli depremi baz alınarak yapılan tasarımların analizlerinin değerlendirilmesi sonucunda elde edilen verilen verileri içermektedir.

Yapı 3 çekirdeği de betonarme olduğunda yapının tasarım ömrü boyunca yorulmalarda düşünüldüğünde güçlendirmeye yada süneklilikten taviz vererek daha rijit yapı inşa edilmesi gerekmektedir. Bu da yönetmelik gereğince mümkün olmamakla beraber yapı deprem gibi doğal afet koşulları altında zayıf düşebilmektedir. Bu da hem mühendislik açısından uygun değildir ve hemde maddi manevi kayıplar önlemek açısından mümkün değildir. Yapının inşa süresi ve kullanılan malzemeler ve yapının kullanım ömrü boyunca yapılan bakım ve onarım maliyetler ve yapımlar düşünüldüğünde yapının maliyeti çelik-betonarme yapıdan daha fazla olmaktadır.

Çelik-betonarme inşa edilmiş yapı rijitlikten taviz vermeden sünekliliği artırılabilen yapı olup deprem gibi yapıyı ciddi zorlayan ve zarar verebilen doğal afet ve yangın gibi yapıyı ciddi seviyede kullanımı etkileyen doğal afetler düşünüldüğünde çelik malzemenin şuan ki teknolojilerle yangına karşı dayanımı artırılabilen yapı olup depreme karşı da dayanımı artırıldığında yapı aynı temelden yükseltikçe orta çekirdek haricindeki sağ ve sol çekirdekler çelik ile inşa edildiğinde yapının inşa süresi ile bakım ve onarım maliyetleri düşünüldüğünde ekonomik açıdan daha uygundur.

Bu yüzden de tasarım anlamında ilk maliyet göz önüne alındığında daha uygun olan betonarme yapılara göre tasarım ve kullanım ömrü ile düşünüldüğünde çelik-betonarme yapılar daha az bakıma ihtiyaç duyup daha fazla kullanım ömrü verebilmektedir. Mesela 3 çekirdeği de betonarme inşa edilmiş yapı; deprem anında yapının rijitliği artırılmış olduğundan yapı depreme karşı direnç göstermek isteyecektir

ve bu da yapının daha çok zarar görmesi ve kullanılabilirliğinin daha hızlı sönmesi anlamına geleceğinden yapının sünekliliğinden taviz verilmemesi gerektiğini söyleyen yönetmeliklerimiz bu konuda mühendislik ve tasarım açısından sınırlarla belirlenmiş çerçeveler içerisinde kalınması gerektiğini belirtmektedir. Eğer yapı rijitliği güvenlik sınırı içerisinde inşa edildikten sonra sünekliliği artırılabilirliğinden dolayı sismik izolatör gibi görev göreceğinden kullanım ömrü uzatılıp insan sağlığını da göz önünde bulundurduğunda yapı tasarım ömrü daha çok kullanıcı dostu, dönüşebilirliğe ve geleceğe dönük inşa edilmesi konusunda daha tercih edilebilmektedir.

2.1 Tasarım Gereksinimler Ve Süreçleri

Bir yapının asıl tasarım gereksinimi, hedefleri kullanım amacını ve tasarımının performansını yerine getirmek ve onun oluşturulduğu ihtiyaçlarını karşılaması amaçlanır. Yapı, binalarda olduğu gibi alanı çevreleyen çevreye karşı barınak ve koruma sağlayabilir; ya da köprüler gibi insanların ulaşımıyla alakalı her alanda malzemelere erişimi de kolaylaştırmıştır. Etkinliğinin memnuniyeti, yapının yapısal ve diğer gereksinimlere uygun olup olmadığına bağlıdır. Yapısal gereksinimler, yapının üzerinde hareket eden kuvvetleri ve yükleri nasıl direneceği ve aktaracağı ile ilgilidir. Birincil yapısal gereksinimi güvenlik ve yapısal mühendis ilk dikkate tasarım ömrü içinde başarısız olmayacak bir yapı üretmek için, ya da hangi başarısızlık sonuçlanmayacak yapı tasarımı önemlidir. Önemli olan bir durumda yapının diğer gereksinimlerinin de yapı kullanımından dolayı sorun oluşturmamalı ve yapı kullanılabilir durumda olmalıdır (Koiter 1959). Diğer tasarım gereksinimleri ekonomi ve uyum olanlar içerir. Hem ilk maliyet ve bakım maliyetini içeren yapının maliyeti, genellikle sahibi için büyük önem taşımaktadır ve ekonominin gereksinimi genellikle yapının tasarımı üzerinde önemli bir etkiye sahiptir. Yapının maliyeti, sadece kullanılan malzemelerin türü ve miktarı tarafından değil, aynı zamanda imalat ve inşa etme yöntemleriyle de etkilenir. Tasarımcı bu nedenle yapının taşıyıcının boyutları yanı sıra inşaat yöntemleri dikkatli önem vermesi gerekir. Yapı içindeki uyum gereksinimleri, yük direnci ve transfer sistemi (yapısal sistem), mimari sistem, mekanik ve elektrik dahil olmak üzere yapının farklı sistemleri arasındaki ilişkilerden etkilenir. Mekanik ve elektrik sistemleri ve yapısal kullanım için gerekli fonksiyonel sistemler. Yapının servis edilebilirliği, sistemler arasında genellikle doğrudan uyum veya eksikliği ile etkilenir. Yapı aynı zamanda çevre

ile uyumlu olmalıdır ve toplum veya fiziksel çevresi ile olumsuz tepki oluşturmamalı ve olmamalıdır. Tasarım genel amacı, tasarım gereksinimlerini karşılayacak bir yapı icat etmektir. Böylece yapısal mühendis direnecek ve güç ve yükleri yeterli güvenlik ile hareket edecek bir yapısal sistem icat etmek istiyor. Çelik inşa edilen yapılar sadece sanayi ya da hastane tarzı yapıların dışında ömrü belirlenmiş olan konut yapılarında da kullanımına başlanmıştır. Endüstriyel ürünler ve endüstri devrimi sonrasında inşaat ve inşaat malzemeleri teknolojisinde gelişimler olmuştur. İnşaat malzemelerindeki teknolojinin gelişmesiyle malzemelerin güvenilirlikleri ve maliyetini azaltmaya başarılı olmuş ve tercih sebebi olmuştur. Yapının belli başlı taşıyıcı özelliği olan temel, kolon ve döşeme gibi ve yardımcı özelliği olan perde gibi taşıyıcılığı olan yapıların inşasında da çelik kullanımına başlanmıştır. Doğal afetlerin getirdiği birçok etken vardır bunlardan ilk akla gelenleri toprak kayması, yapı hasarları, göçmeler veya meydana gelen can kayıpları sonuçlar elde edilebildiğinden hem yapının taşıyıcı özelliğini önde tutup hem de maliyetten tasarruf edip hem de gelişmiş teknolojiyle fiziksel ve etkileşimli ortamına göre malzeme özelliğini tercih ederek yapının inşası için gerekli olan koşulları sağlanabilmektedir.

Yapı inşa esnasında tasarımın adım adım atılacak durumları şöyle özetleyebiliriz;

Çizelge 2.1: Yapım Aşamaları

Yer ve çevresindeki sorunların tanımı ihtiyaç, kısıtlamalar ve hedefler
Alternatif durum, koşul ve sistemler (yapısal, genel ve diğer sistemler)
Öntasarım esnasında yapısal tasarım, yükler, tasarım analiz ve diğer projeler (mimari, elektrik ve mekanik)
Öndeğerlendirme yapının güvenliği, ekonomi, projelerin harmonisi
Proje düzenlemeri
Son Tasarım aşamasında yapının tasarımının ve diğer tasarımlar sonlanması
Son değerlendirme ve şartnamelere uygunluğu
Yürütme aşamasında ihale, inşa ve denetim, belgelerin tamamlanması
Tasarım özetleme aşamasında bilgi, deneyim, hayal gücü, sezgi ve yaratıcılık
Ön analiz aşamasında yaklaşım, yükler ve davranış
Tasarım kriterleri ve kodları
Analiz, yükler ve davranış

İnşaat mühendisi, mimar, elektrik ve makina mühendisi tasarımlarında kişisel gelişimlerini kullanırlar. Alternatif genel sistemlerin bir dizi ve hedefleri karşılamak için görünen ilişkili yapısal sistemleri icat etmektir. Bunu yaparken, tasarımcı kişisel bilgi ve deneyimi kullanabilir veya diğerlerinden veriler toplanıp derlenebilir. Tasarımcılar kendi hayal, sezgi ve yaratıcılık veya bunların bir kombinasyonu kullanabilir. Yapı inşasına geçmeden analizler yapıp tekrardan kullanım amacına ve tasarım ihtiyaçlarını karşılayacak şekilde yüklerin uygulanabilirliğine bakılmalıdır ve yaklaşık varsayımlarla yapıp analizler sonucunda ve gelen düşey ve yatay kuvvetlere karşı ve hareket eden kuvvetlerde de hesaplanıp yeniden son tasarıma ulaşılır. Bu aşamada yapının bir ön tasarımı tamamlanmıştır, ancak yapısal davranışlar hakkında yapılan yaklaşık varsayımlar nedeniyle, tasarımı kontrol etmek gerekir. İlk adımlar, yükleri yeniden hesaplamak ve tasarlanmış yapıyı yeniden analiz etmek ve bunlar, ön analiz için mümkün veya uygun olandan daha hassasiyetle gerçekleştirilir. Yapının performansı, yapısal gereksinimlerle ilgili olarak değerlendirilir. Yapının performansı, yapısal gereksinimlerle ilgili olarak değerlendirilir ve taşıyıcı elemanların boyutlarında ve konumlarında yapılan değişiklikler karar verilir. Bu değişiklikler yapının daha fazla yeniden analizini ve yeniden orantısız olmasını gerektirebilir ve başka bir değişiklik gerekene kadar bu döngü tekrarlanabilir. Alternatif olarak, orijinal yapısal sistemi değiştirmek ve tatmin edici bir yapı elde edilinceye kadar yapısal tasarım sürecini tekrarlamak gerekebilir. Alternatif genel sistemler daha sonra hizmet etme, ekonomi ve uyum açısından değerlendirilir ve yukarıda belitildiği listedeki gibi sırasıyla son sistem seçilmiş olur. Bu son genel sistem, tasarım sonlandırılmış önce değiştirilebilir. Ayrıntılı çizimler ve Teknik özellikler daha sonra hazırlanabilir ve İnşaat için ihaleler çağrılabilir ve izin verebilir ve yapı inşa edilebilir. Daha fazla değişiklik, yapılan ihaleler sonucunda veya inşaat sırasında keşfedilen öngörülemeyen koşullara bağlı olarak yapılmalıdır.

Yapının tasarımında seçilmiş olan çelik standartlara uygun şekilde ve yapısal gereksinimleri ile ilgili olarak, çelik yapıların yapısal davranışları ve davranışları ve onları orantılı yöntemleri arasındaki ilişkileri ile ilgilidir. Bu ilişkiyi standartlara göre belirlenir. Ülkemizde kullanılan standart TS EN 1994-1-1 (Çelik ve beton karma yapıların tasarımı) kuralları üzerinden de değerlendirilir ama bu aşamada yardımcı diğer yönetmeliklerimizi de kullanılmamız gerekecektir. Ülkemizde yardımcı diğer

yönetmeliklerimizden bir tanesi de TBDY (Türkiye Bina Deprem Yönetmeliği – 2018) bu yönetmelikten yapımızın tasarımında kullandığımız gibi malzeme ve yüklerimize göre kullanacağımız malzemelerin maruz kalacağı yüklerin tasarımını çözümleriz.

Tasarım 6 katlı yapıdan oluşmakta olup yapı 3 farklı program üzerinden tasarım yapılmıştır. Sap2000, ProBina ve Seismo Struct programları sonlu eleman kullanılarak tasarlanmıştır.

CSI (Computer and Structures Inc.) Ameri şirketi tarafından yayınlanmakta olan Sap2000 programı statik tabanlı sonlu elemanlar yöntemiyle tasarlanmış çoğunlukla inşaat mühendislerinin kullandığı programdır. Bu program statik tabanlı olduğundan dolayı yapıya ait her elemanı (kolon, kiriş, döşeme ve malzeme) gibi herşeyin ağırlığını, dayanımlarını, sünekliliklerini ve elastik özelliklerini elle girilerek bazı bilgiler (yükler, deprem ve yönetmelikler) gibi programa verilir bu sayede program program paket program haline dönüştürülmüş olunur. Ayrıca program yaygınlığının yanı sıra Autocad programından çizilmiş olan mimari projeleri içeri aktar özelliği ile yada program içi tasarım Autocad'e aktar özelliği ile bilgi ve veri girdi çıktısı kolay olan bir programdır. Ayrıca program ile spa ile yapı sistemlerinin çözümlenmesi, çubuk (frame), 3 boyutlu (solid), dönel simetrik ve kabuk gibi elemanlarının tanımlamaları üzerinden malzeme ve kesit bilgileriyle yükleme ve yük kombinasyonları ile statik ve dinamik analizleri yapmak çerçeve tipi yada kafes tipi yapılarda deprem kuvvetlerinin çözümlenmek mümkündür.

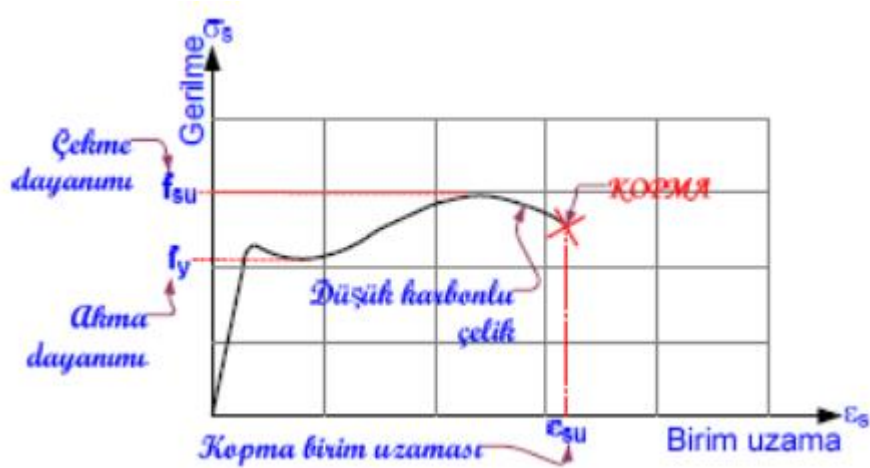
Seismo struct programı deprem üzerine tasarlanmış sonlu elemanlar yöntemini kullanan deprem üzerine odaklanmış

Probina programı paket program olarak adlandırılan her türlü bilgi ve tanımlama program içerisinde mevcut olan programdır. Program statik tabanıyla sonlu elemanlar yöntemi ile çözümlenme yapmakta olup bütün yük, malzeme vb hertürlü bilgiyi program içerisinden alarak kullanılabilenmekte olduğundan program kullanışlığı yüksek olup gerekli diğer etki ve faktör parametreleri belirtmek mümkün olamamaktadır. Bu çalışma içerisinde diğer parametrelere ihtiyaç kalmadığından çözümlenme konusunda sorun olmamıştır.

2.2 Malzeme Seçimi

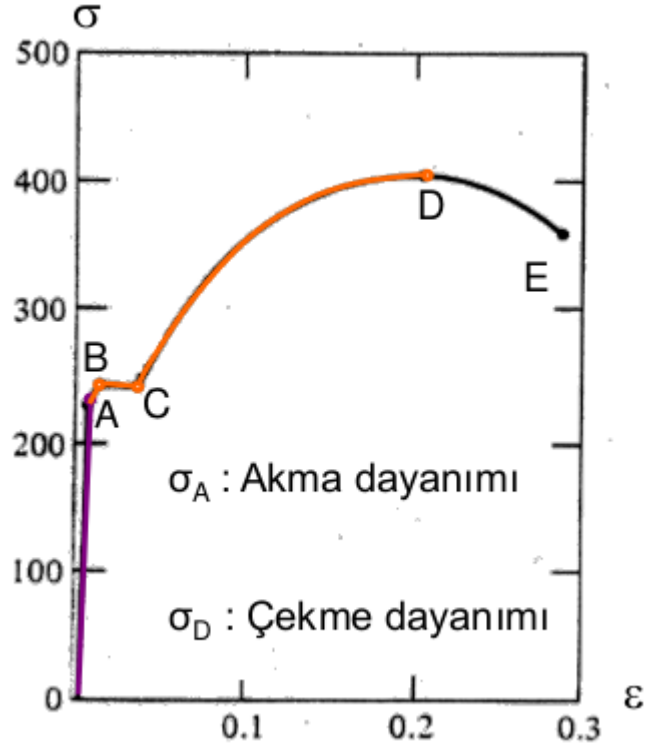
Statik yük altında çoğu yapısal çeliğin önemli mekanik özellikleri Şekil.2.1 gösterilen idealleştirilmiş çekme stres-gerinim diyagramında belirtilir. Çelik malzemesinin gerilme/birim uzama ($G = P/A$) eğrisinden görülebileceği gibi 3

bölümde inceleyebiliriz ve bunlarıda Elastik, Elasto-Plastik ve Kopma olarak adlandırabiliriz. Sonraki bölümlerde de ayrıntısına girilecek süneklik konusunda Elastisite modülünün önemi fazladır. Elastisite Modüsü (E) değerleri 200.000 210.000 MPa aralığında farklılık gösterir ve yaklaşık 200 000 MPa değeri genellikle kabul edilir. Çelik bu doğrusal aralıkta elastik seviyesinde kalır ve üzerindeki yükü malzemede değişiklik olmadan boşaltmayı mükemmel kurtarır. Doğrusal elastik davranışların sınırı genellikle verimli G ve karşılık gelen verimli gerilme ile yakından hesaplanabilir. Bu sınırın ötesinde G geriliminin en son elasto-plastik seviyesine ulaşılan kadar ki çelik, gerilmenin herhangi bir artış olmadan elastik şekilde davranarak hareket eder ve önemlidir çünkü çelik süneklik hesaplanır. Şekil 2.2 görülebileceği gibi A noktasından sonra plastik özelliklerine sahip olurken A-C Akma bölgesi olarak isimlendirilir ve bu bölgede malzemede gerilme artmaksızın şekil değiştirme görülür. C-D arası Pekleşme bölgesi olarak isimlendirilir ve Şekilde değişiklik oluşmaya devam etmektedir ve artık plastik bölgede olduğundan malzemede gözle görülen değişikliklere sebep olacaktır. D-E bölgesi artık malzemenin aşırı uzama yaparak boyun verme diyebileceğimiz bir bölgedir ve önceki bölgelerden çok farklıdır ve bu bölgenin sonunda kopma olacaktır. Kopma anındaki en başta kabul edilen MegaPascal dayanımının gerçekte ne kadar olduğuyla alakalı deney sonucu elde ettiğimiz deneydir.



Şekil 2.1: Çelik Çekme Grafiği

(<https://www.medeniyetmuhendisleri.com/index.php?topic=2648.0> – 17/09/2019)



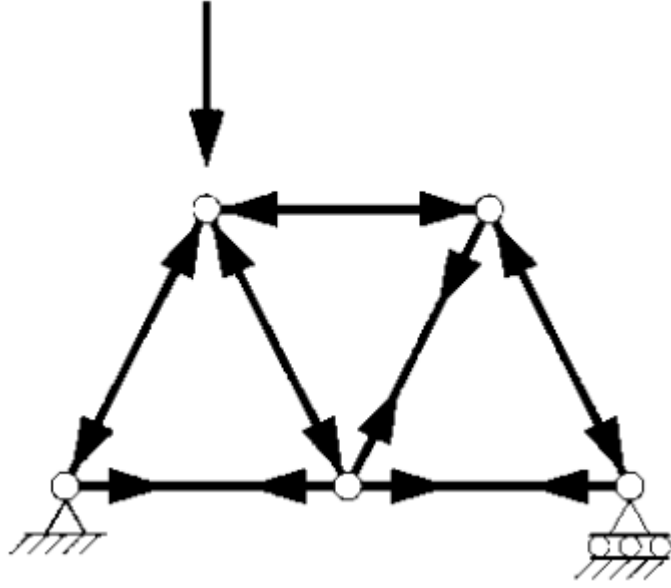
Şekil 2.2: Çelik Çekme Grafiği-2

(<https://www.medeniyetmuhendisleri.com/index.php?topic=2648.0> – 17/09/2019)

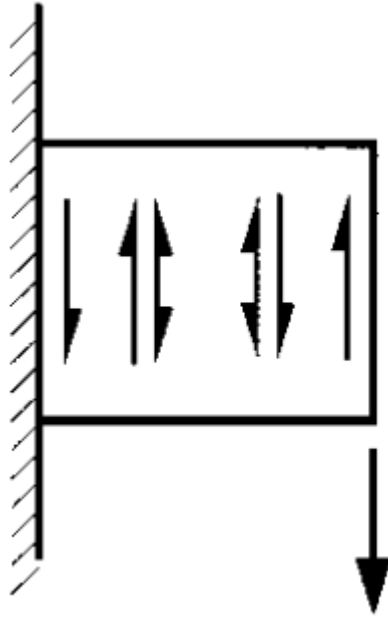
Tasarım amacıyla, her farklı çelik sınıflandırması için minimum verim gerilimi tanımlanır. Belirli bir çeliğin asgari verim gerilimi, bir dizi standart gerilim testinin sonuçlarından belirlenir.

2.3 Yapının Davranışı

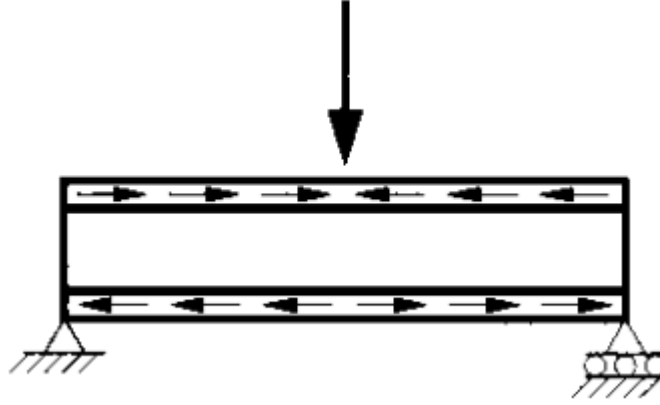
Bir yapının davranışı, taşıyıcı elemanları ve bağlantılarının yük aktarımı eylemine bağlıdır. Bunu tasarım esnasında zaten açıklandı.



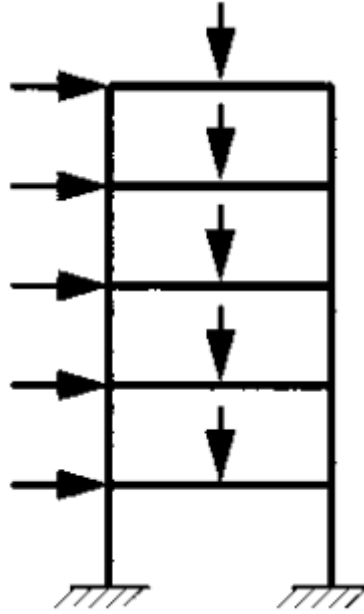
Şekil 2.3: Yapının Davranışı



Şekil 2.4: Yapının Davranışı



Şekil 2.5: Yapının Davranışı



Şekil 2.6: Yapının Davranışı

Çelik Lamine Kauçuk Mesnet (SLR) elasto-plastik olduğundan daha fazla dayanım göstermiştir. Ayrıca diğer yaygın kullanımı olan Kauçuk Mesnet(RB)

Yüksek Sönümlü Doğal Kauçuk Mesnet(HDNR) malzemelerden kullanımı yaygın ve dayanımları yüksektir. Bu deneylerle kanıtlanmış akademik çalışmalardan elde edilmiştir(G.Gedge. Structural uses of stainless steel). Bu malzemeleri mesnet göreviyle kullanılmış izolatör görevli yapılar mevcuttur. Ama kullanımı maliyet sebebiyle daha da yaygınlaşmadığından konut gibi yapılarda kullanımı artmamıştır. Gösterildiği gibi eklem yükleme ile üçgen yapılarda olduğu gibi, aksenal gerginlik veya sıkıştırma neredeyse tamamen olabilir. Alternatif olarak, Üyeler bükme ve kesme eylemleri ile aktarılan enine yükleri destekleyebilir. Genellikle bükme eylemi, kirişler ve çok sayıda tek katlı sert çerçeveler Şekillerde görüldüğü gibi kolonlardan üyelerden

oluşan yapılarda hakim olurken, kesme iki boyutlu plaka yapısında daha önemli hale gelir. Birçok yapının üyeleri, çok katlı binalarda olduğu gibi Aksiyel kuvvetlere ve enine yüklerine tabi tutulur. Bir yapının üyelerinin yük aktarma eylemi, geometrik mizanpaj ve bağlantı detayları ve yükleme düzenlemesi de dahil olmak üzere yapının düzenlenmesine bağlıdır. Bazı yapılarda, yükleme ve bağlantı üyeleri etkili bir şekilde bağımsızdır. Örneğin, eklem yükleri ile üçgen yapılarda, herhangi bir esneme etkileri sekonder, ve Üyeler pin-eklemli gibi hareket kabul edilebilir, basit esnek bağlantıları olan dikdörtgen çerçevelerde ise giriş ve sütunlar arasında an transferleri olabilir yok sayılır. Bu gibi durumlarda, yapının yanıtı doğrudan bireysel üye yanıtlarından elde edilir. Ancak daha genel olarak, Üyeler arasında etkileşimler olacak ve yapı davranışları, karşılaştırılarak görülebilir, bir üyenin genel davranışından farklı değildir. Böylece, bir çelik yapı hizmet yükleri altında elastik davranır varsayalım geleneksel olmuştur. Bu varsayım, kalıntı stresleri ve stres konsantrasyonları nedeniyle yerel erken verim yoksayar, ancak bu genellikle ciddi değildir. Tamamen esneme yapıları ve hafifçe yüklenmiş sıkıştırma üyeleri ile tamamen Aksiyel yapılar, doğrusal gibi davranır. Ancak, hem esnek hem de eksenel eylemlere sahip yapılar, hizmet yüklerinin yakınında bile doğrusal olmayan davranır. Bu, üyelerinin geometrik olarak doğrusal olmayan davranışının bir sonucudur. Çoğu çelik yapıları, onların nihai yükleri yakın doğrusal olmayan davranır, onlar erken kırılma kırık nedeniyle başarısız olmadıkça, yorgunluk, ya da yerel belverme. Bu doğrusal olmayan davranışlar, malzeme elde etme veya taşıyıcı sistem veya çerçeve belverme nedeniyle yapılır. Aksiyel yapılarda, hata bazı tension üyelerinin veya bazı sıkıştırma üyelerinin veya çerçevenin ya da her ikisinin de belverme gerektirebilir. Esnek yapılarda başarısızlık, yapının bir çöküş mekanizması oluşturmasında yeterli sayıda konumda meydana gelen tam plastisite ile ilişkilidir. Eksenel ve esneme eylemlerine sahip yapılarda, verimli ve belverme arasında bir etkileşim vardır ve başarısızlık yükü genellikle belirlemek zordur. Esnek ve nihai davranış arasındaki şekilde gösterilen geçişler, bireysel elemanlar tamamen plastik veya toka haline gelene kadar genellikle doğrusal olmayan adımlarla gerçekleşir.

2.4 Yapıya Ait Yükler

Çelik yapı yada çelik ve betonarme yapı yada betonarme yapılarda yada diğer taşıyıcı malzemeli yapılarda hesaplamaların mantığı ve düşüncesi aynı diyebiliriz. Yapının bağlı kaldığı koşullar göz önüne alındığında yapının yüklerinin hesabını yapmakta zorlanılmaz. Koşullar ve yönetmeliklerimiz dikkate alındığında rüzgar, kar ve deprem gibi dış yükler haricinde dinamik yüklerimizde mevcuttur. Bu yapının kullanım ömrü

ve kullanım şekline göre seçilecektir. Yük hesapları yönetmeliğe bağlı ve belirtildiğine göre hesaplamalar yapılır bunun dışında mühendislerin inisiyatif sınırları içerisinde hesaplamalar yapılır. Yönetmelikte olmayan yükler yada yük sınırları içerisinde bölgeye ait yönetmelikte olmayan yükler inşaat mühendislerinin ve belediye sorumluları inisiyatif içerisinde yükler tanımlarak belirlenir ve buna göre yük hesaplaması işlemi yapının kullanım amacına göre yapılabilir. Yapı tasarımında mahallere göre öngörülen düşey ve yatay yükler belirlenmiştir. Yük kombinasyonları TS-500 ve Türkiye Bina Deprem Yönetmeliği (2018) içerisinde deprem etkisi altında binaların tasarımı için yönetmelik içerisinde bulunan yük kombinasyonlarımızı alınmıştır. Bu yük kombinasyonları ve yapının geleceği esnasında değişiklik gösteren faaliyet yada kullanım amacı değiştiğinde destekleyici güvende kalacak yük kombinasyonları tanımlanırken yönetmeliklerimiz referans alınırken yapının rijitlik ve süneklilik kurallarımızdan vazgeçemeyeceğimizden tasarım ve dizayn önemlidir. Yapı sürekli aynı yükler altında deformasyon gösterirken anlık yükler yada yük aşımı durumunda yapının güvende kalacak kombinasyonlar içerisinde desteklese dahi yapı zorlanmaya ve yorulmaya başlayacaktır. Bu da yapının kullanım ömrünü azaltacağını biliyoruz.

2.5 Deprem

Doğal afetlerin etkililerinden biri olan deprem, Türkiye'nin coğrafi konumu gereği fay hatları üzerinde olduğu ve bunların yerlerin ve büyüklüğü birden çoğu belirlenmiştir. Depremlerin yenilenme süreleri ve şiddetlerine göre yapının kaç deprem geçireceği yada ne kadar şiddetli depreme maruz kalacağı konusunda tahminler yürütülmektedir. Bunların neredeyse hepsi DBYBHY (Deprem Bölgelerinde Yapılacak Binalar Hakkında Yönetmelik – 2007) ve TBDY (Türkiye Bina Deprem Yönetmeliği – 2018) yönetmeliklerimizde ayrıntılı bir şekilde belirtilmiştir. Deprem hareketiyle alakalı çözümleyeceğimiz örnekle de anlaşılacağı gibi DD-1 (2475 yıl), DD-2 (475 yıl), DD-3 (72 yıl) ve DD-4 (50 yıl) olarak depremin hareket yüzeylerini ayrıntısıyla beraber inceleyebiliriz(D.Özhendekci. Çelik Yapılar I-II Ders Notları YTÜ İnşaat Müh. Böl., İstanbul.).

Çözümlemelerde kullanılmış olan programlarda bunlar göz önünde bulundurulmuş olup yardımcı yapı elemanları ve güçlendirilmiş malzemeler kullanımı seçilmiştir(Yazgan, U., R. Oyguç, M.E. Ergüven ve Z. Celep, (2016). "Seismic performance of buildings during 2011 Van earthquakes and rebuilding efforts", Earthquake Engineering and Engineering Vibration 15(3):591-606).

Yardımcı yapı elemanları olan geçen toprak güçlendirilmesi veya zeminden gelecek olan yanal yükler altında yapı zarar görmesin diye kullanılan Fore Kazı sistemi elemanların yardımıyla yapı güçlendirilebilir. Kazıktan zemine yük aktarma mekanizması oldukça karmaşıktır çünkü toprak altından birden çok farklı yönden aynı anda yük geldiğinden dolayı fonksiyonlu formülleri üzerinden çözümlenebilir. Formülde gösterilecek olan "L" kazık uzunluğu boyunca yük dağılımının irdelenerek çözümlenebilir. Kazık üzerinde "Qw" zemin yüzeyinden gelen yükür. By yük derinlere indikçe tabii ki azalmaktadır. Toprak içerisinde kısmı kazık gövdesi boyunca "Qws" ve kazık ucunun altında kalan zemin tarafından karşılanan "Qwp"dir. Yük aktarma mekanizması toplam yükün ne kadarlık kısmının "yüzey sürtünmesi" ne kadarlık kısmının "uç mukavemeti" ile karşılanacağını belirlemektedir(Taşkın, G., Ersoy, O.K., Kamaşak, M.E., 2015). Eğer herhangi bir "z" derinliğinde kazık şaftı tarafından taşınacak [Qz] yükünün belirlenmesi için ölçümler yapılırsa "Qws" yükünün şaft boyunca dağılımı aşağıdaki formülde belirtilmiştir. Herhangi bir "z" derinliğinde birim şaft alanındaki yüzeysel sürtünme direnimi "fz" ile saplanabilmektedir. Ayrıca tasarım spektrum parametreleriyle AFAD deprem fay hattındaki önemini de anlayabiliriz.

$$f_z = \frac{\Delta \cdot Q_z}{[P] \cdot [\Delta z]} \quad \text{Formül1: Sürtünme Kuvveti Gerilme Formülü}$$

Çizelge 2.2: Tasarım Spektrum parametreleri

	DD-1	DD-2
S_{DS}	1.35 g	0.73g
S_{D1}	0.504 g	0.28
T_A	0.075 sn	0.076 sn
T_B	0.37 sn	0.38 sn
PGA	0.54 sn	0.29 sn

Harita spektral ivme katsayıları S_s ve S_1 , aşağıdaki şekilde tasarım spektral ivme katsayıları S_{DS} ve S_{D1} 'e dönüştürülür (Türkiye Bina Deprem Yönetmeliği 2018).

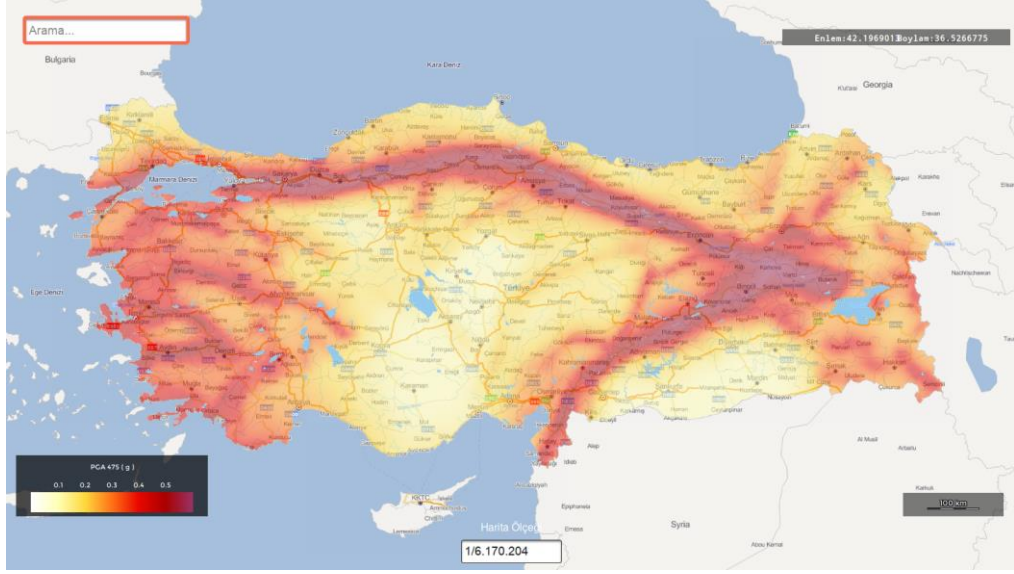
$$S_{DS} = S_s F_s$$

Formül2: Harita spektral ivme katsayıları formülü

$$S_{D1} = S_1 F_1$$

Formül3: Harita spektral ivme katsayıları formülü

Türkiye Bina Deprem Yönetmeliği 2018 göre incelendiğinde 16 adet deprem harita mevcuttur. Bu haritaların içerisinde aynı bölgede olan 2 farklı yapının Deprem yer hareketi düzeyi farklılık gösterebileceğinden yapımızın en sık denk gelme ihtimalimiz olan SS-43, S1-43, PGA-43 ve PGY-43 harita baz alınarak yapılmış olan deprem analizlerimizde DD-4 yer hareketi uygun olarak alınmıştır ve aşağıda göreceğiniz resimde elde edilen haritaya göre Kocaeli depremi uygulanmıştır. Ekte görülebilecek örnekli çözümleme ile anlaşılabilir. Ekte görülebilecek örnekli çözümleme ile anlaşılabilir.

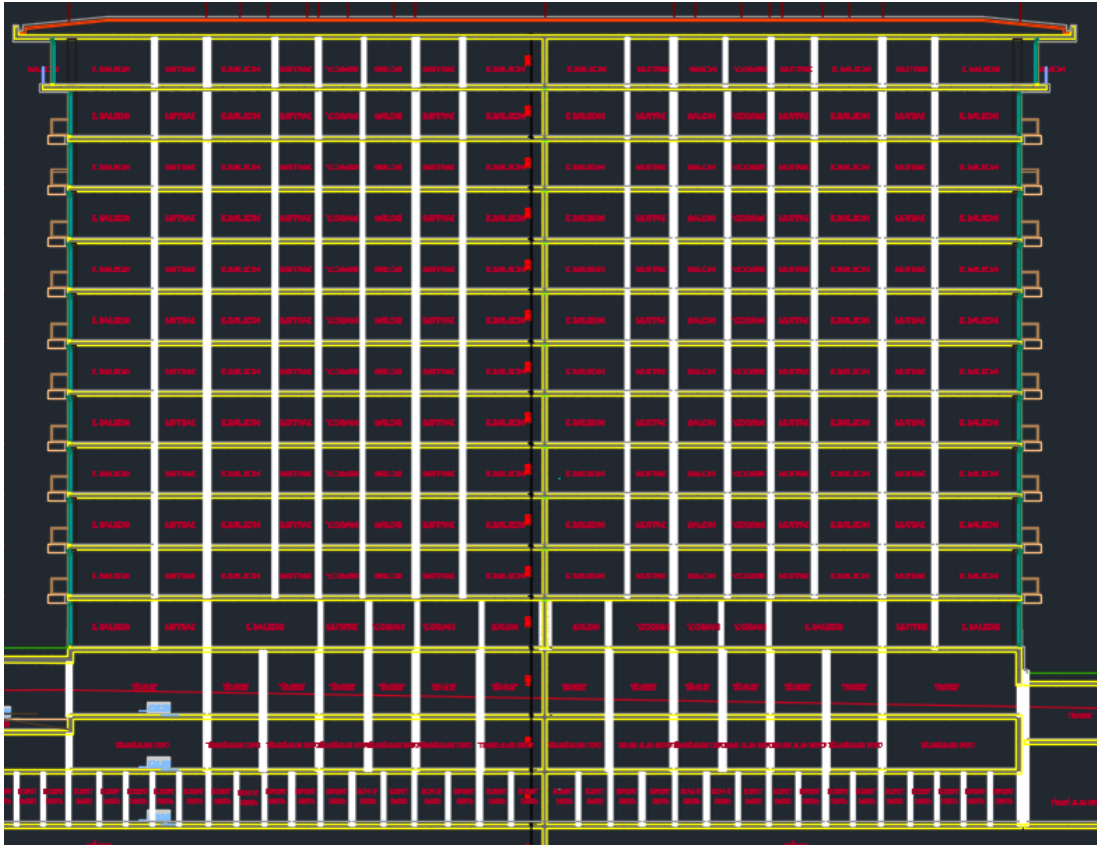


Şekil 2.7: AFAD Deprem Fay Hattı (AFAD, 2018)

3. ANALİZ VE SONUÇLAR

Yapılmış olan yapı analizlerinde 12 katlı betonarme bina Sap2000 programı ile tasarlandıktan sonra yapı kıyaslama ve ölçüm yapılabilmesi için 6.katlı çelik ile güçlendirilmiş betonarme yapı tasarlanmıştır. Çıgıb yapı sonlu elemanlar yöntemleriyle Sap2000 v18, ProBina v18 ve Seismo Struct v7 programı ile yapılmış olan tasarımlar ekran görüntüleri çalışma içerisinde görülebilmektedir.

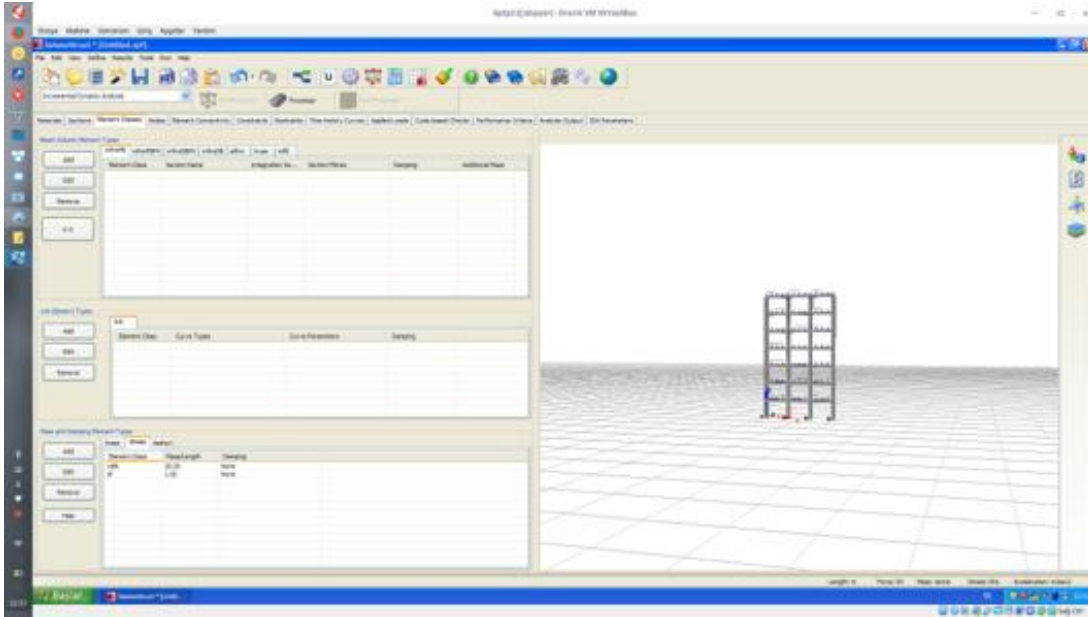
3.1 AutoCad Tasarım Görüntüsü



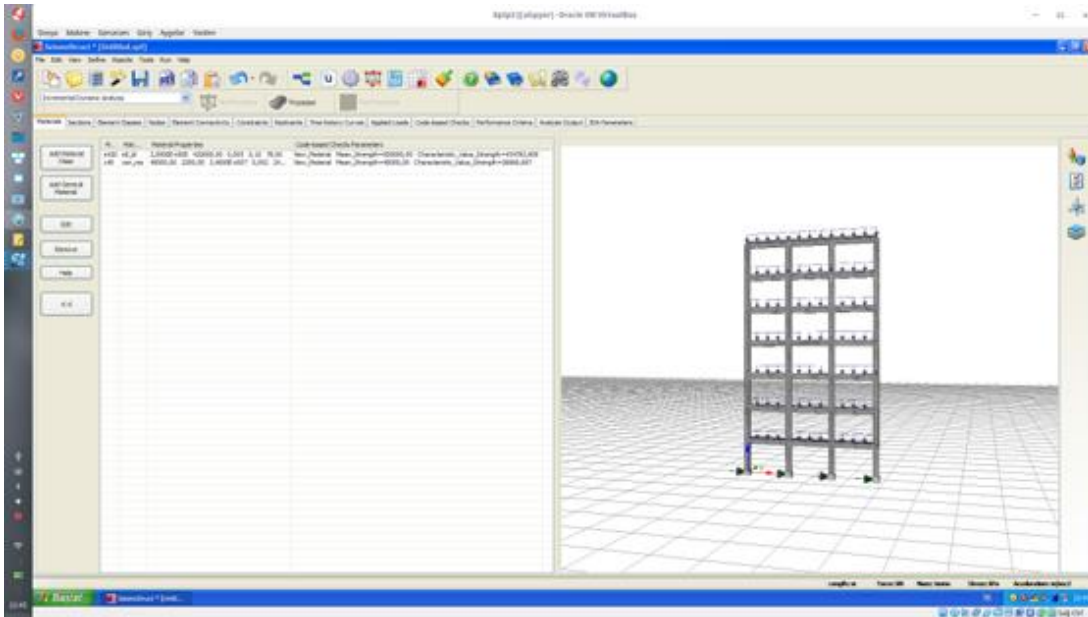
Şekil 3.1: Autocad programından 12 katlı tasarımın yapının ön görünüşü

3.2 Seismo Struct Programı Tasarım Görüntüleri

Programın çalışma şekli ile alakalı 2.1. Tasarım Gereksinimleri ve Süreçleri başlığı altında açıklandığı gibi yapı deprem için dizayn edilmiş analiz programıdır.

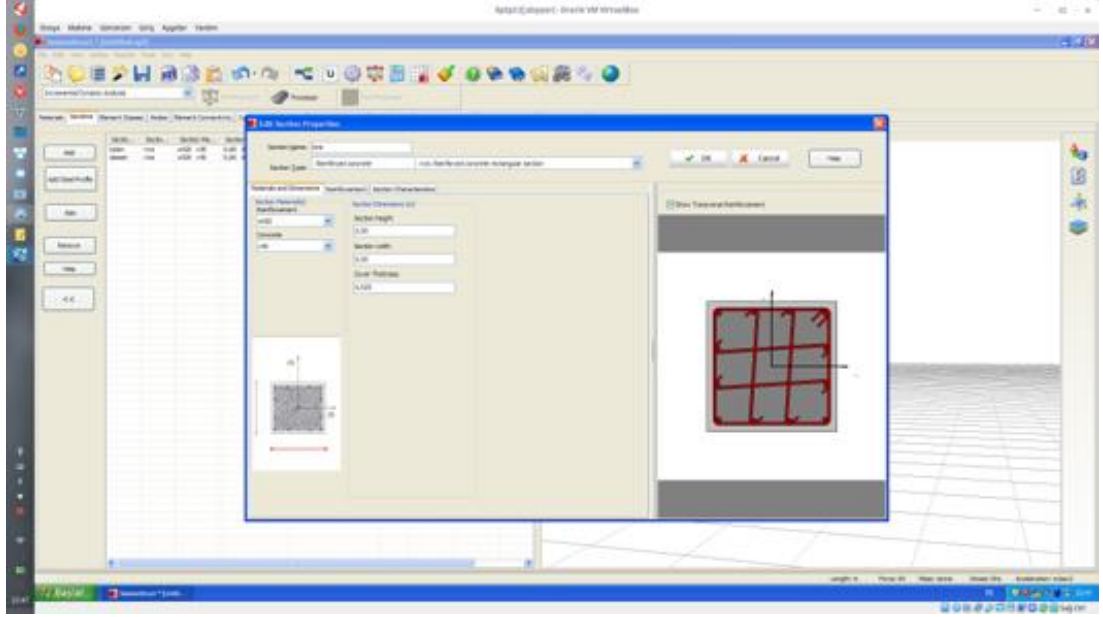


Şekil 3.2: Seismo Struct programından 6 katlı tasarımın yapının ön görünüşü



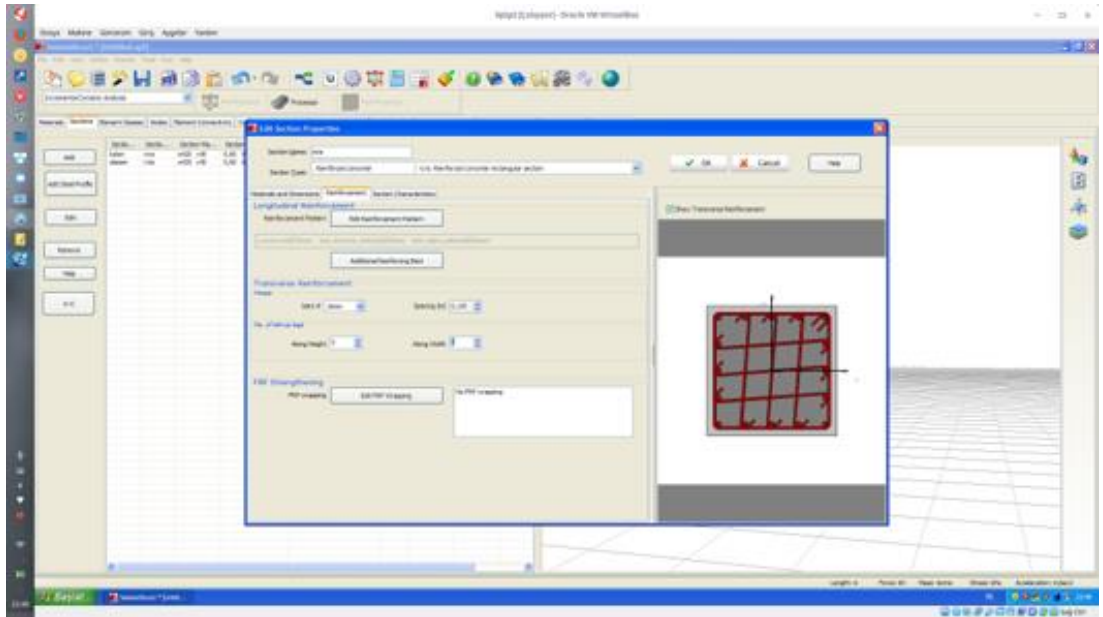
Şekil 3.3: Seismo Struct programından 6 katlı tasarımın yapının çelik ve betonarme bilgilerinin ön görünüşü

Şekilden anlaşıldığı gibi çelik betonarme bilgileri yukarıda belirtildiği içerikle beraber girilmiştir. Çelik H360 ve C40 beton kullanılmıştır ve betonarme içeriği de 8 ad. Çevrenmiş donatı kullanılmıştır.



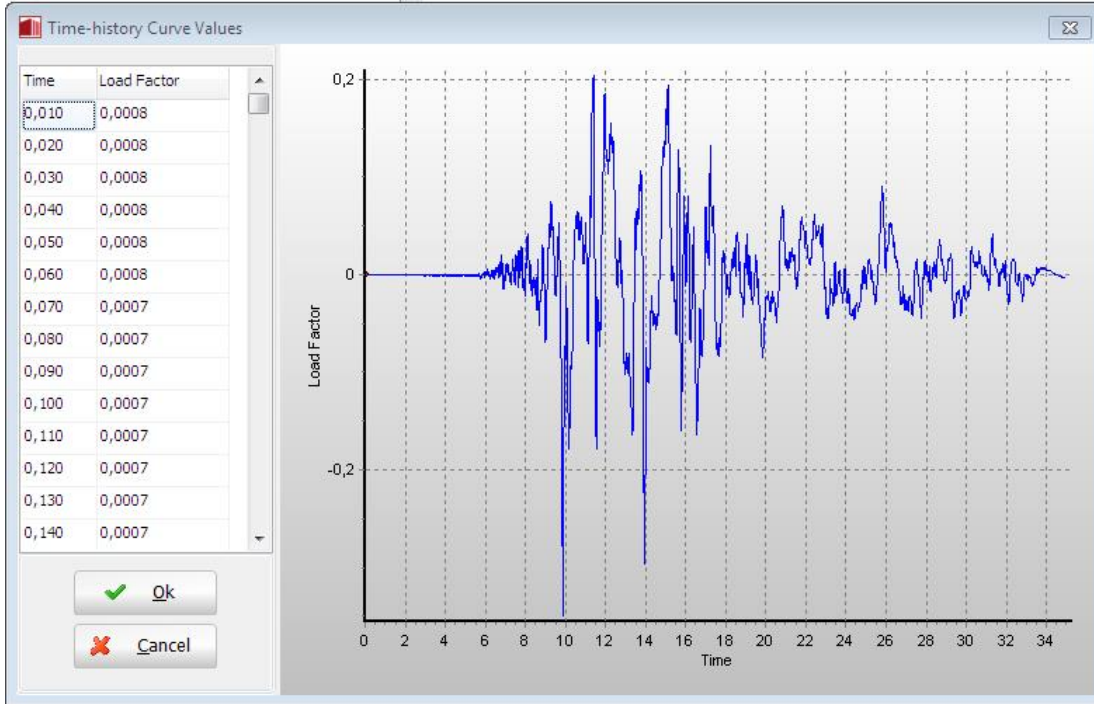
Şekil 3.4: Seismo Struct programından 6 katlı tasarımın yapının donatı kesiti

Donatıların kirişte ve kolonda tasarlanma şekli gösterilmeye çalışılmıştır.



Şekil 3.5: Seismo Struct programından 6 katlı tasarımın yapının donatı kesiti

Betonarme yapının donatı dizilişinin ve kullanılan malzeme içeriği görünmektedir.



Şekil 3.6: Seismo Struct programından Kocaeli depremin Kandilli/İstanbul merkezli deprem hareket yüzeyi

3.3 Seismo Struct Programı Analiz Sonuçları

Bu paket program aracılığıyla betonarme ve çelik birleşimlerini kaynaklı yada blon hesabını ve tasarımını yapmak daha mümkün olduğundan ve hesaplamalarını diğer programlarla entegreli çalıştırabilmek için seçilmişken aynı anda her donatı ve her çelik plakayı en son ne duruma kadar hangi alanda zorlandığını ve en kesidini görmek mümkündür. Ayrıca analiz programında gördüğümüz gibi donatıların ve yapı taşıyıcı malzemelerin ne aşamada olduğu ile alakalı ayrıntılı bilgileri göz gezdirilebilir.

Performance Criteria

Otp No: 977 Time= 9,7600, yield reached. Elm: bmx314. Reinf.
Steel Strain = 0,0025047 - Sec(b)

Otp No: 979 Time= 9,7800, yield reached. Elm: bmx313. Reinf.
Steel Strain = 0,00253492 - Sec(b)

Otp No: 980 Time= 9,7900, yield reached. Elm: bmx314. Reinf.
Steel Strain = 0,00258078 - Sec(a)

Otpt No: 981 Time= 9,8000, yield reached. Elm: bmx313. Reinf.
Steel Strain = 0,00251676 - Sec(a)

Otpt No: 981 Time= 9,8000, yield reached. Elm: bmx315. Reinf.
Steel Strain = 0,00254667 - Sec(b)

Otpt No: 985 Time= 9,8400, yield reached. Elm: bmx214. Reinf.
Steel Strain = 0,00252411 - Sec(a)

Otpt No: 985 Time= 9,8400, yield reached. Elm: bmx214. Reinf.
Steel Strain = 0,00251886 - Sec(b)

Otpt No:1015 Time=10,1400, yield reached. Elm: bmx113. Reinf.
Steel Strain = 0,00257523 - Sec(a)

Otpt No:1017 Time=10,1600, yield reached. Elm: bmx113. Reinf.
Steel Strain = 0,00261245 - Sec(b)

Otpt No:1018 Time=10,1700, yield reached. Elm: bmx213. Reinf.
Steel Strain = 0,00251193 - Sec(a)

Otpt No:1018 Time=10,1700, yield reached. Elm: bmx213. Reinf.
Steel Strain = 0,0025105 - Sec(b)

Otpt No:1019 Time=10,1800, yield reached. Elm: bmx114. Reinf.
Steel Strain = 0,00263529 - Sec(a)

Otpt No:1019 Time=10,1800, yield reached. Elm: bmx312. Reinf.
Steel Strain = 0,0026648 - Sec(b)

Otpt No:1020 Time=10,1900, yield reached. Elm: bmx112. Reinf.
Steel Strain = 0,00263633 - Sec(a)

Otpt No:1020 Time=10,1900, yield reached. Elm: bmx114. Reinf.
Steel Strain = 0,00256303 - Sec(b)

Otpt No:1020 Time=10,1900, yield reached. Elm: bmx116. Reinf.
Steel Strain = 0,00257013 - Sec(a)

Otpt No:1021 Time=10,2000, yield reached. Elm: bmx112. Reinf.
Steel Strain = 0,00257737 - Sec(b)

Otpt No:1021 Time=10,2000, yield reached. Elm: bmx212. Reinf.
Steel Strain = 0,00259709 - Sec(b)

Otpt No:1021 Time=10,2000, yield reached. Elm: bmx115. Reinf.
Steel Strain = 0,00281145 - Sec(a)

Otpt No:1021 Time=10,2000, yield reached. Elm: bmx115. Reinf.
Steel Strain = 0,00257682 - Sec(b)

Otpt No:1021 Time=10,2000, yield reached. Elm: bmx116. Reinf.
Steel Strain = 0,00250219 - Sec(b)

Otpt No:1022 Time=10,2100, yield reached. Elm: bmx212. Reinf.
Steel Strain = 0,00253335 - Sec(a)

Otpt No:1022 Time=10,2100, yield reached. Elm: bmx312. Reinf.
Steel Strain = 0,00258611 - Sec(a)

Otpt No:1022 Time=10,2100, yield reached. Elm: bmx117. Reinf.
Steel Strain = 0,00252205 - Sec(a)

Otpt No:1023 Time=10,2200, yield reached. Elm: bmx215. Reinf.
Steel Strain = 0,0026213 - Sec(a)

Otpt No:1023 Time=10,2200, yield reached. Elm: bmx215. Reinf.
Steel Strain = 0,00261244 - Sec(b)

Otpt No:1025 Time=10,2400, yield reached. Elm: bmx216. Reinf.
Steel Strain = 0,0026727 - Sec(a)

Otpt No:1025 Time=10,2400, yield reached. Elm: bmx216. Reinf.
Steel Strain = 0,00266443 - Sec(b)

Otpt No:1025 Time=10,2400, yield reached. Elm: bmx117. Reinf.
Steel Strain = 0,00251398 - Sec(b)

Otpt No:1026 Time=10,2500, yield reached. Elm: bmx315. Reinf.
Steel Strain = 0,00273837 - Sec(a)

Otpt No:1029 Time=10,2800, yield reached. Elm: bmx316. Reinf.
Steel Strain = 0,00251882 - Sec(b)

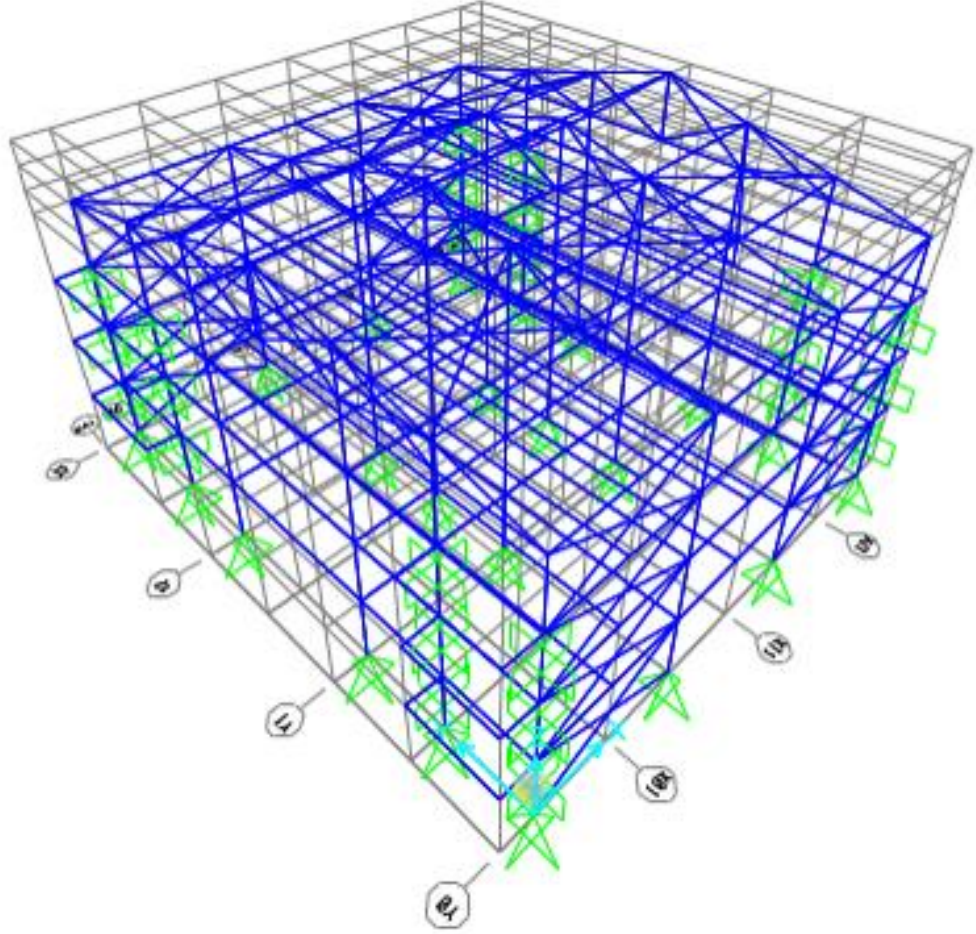
Otpt No:1033 Time=10,3200, yield reached. Elm: bmx316. Reinf.
Steel Strain = 0,00252606 - Sec(a)

Otpt No:1037 Time=10,3600, yield reached. Elm: col311. Reinf.
Steel Strain = 0,00260851 - Sec(a)

Otpt No:1041 Time=10,4000, yield reached. Elm: col211. Reinf.
Steel Strain = 0,00266246 - Sec(a)

3.4 Sap2000 Programı Tasarım Görüntüleri

Programın çalışma şekli ile alakalı 2.1. Tasarım Gereksinimleri ve Süreçleri başlığı altında açıklandığı gibi hertürlü yapının tasarımı yapılabilmekte olup, analiz için gerekli hertürlü bilgi elle gerekli parametreler içinde yapılması gerekmektedir. Program paket program olmadığından çok tercih edilmesede inşaat mühendisleri için yapılmış bu programın dünya çapında yaygındır.



Şekil 3.7: Sap2000 programından 6 katlı tasarımın yapının ön görünüşü

3.5 Sap2000 Programı Analiz Sonuçları

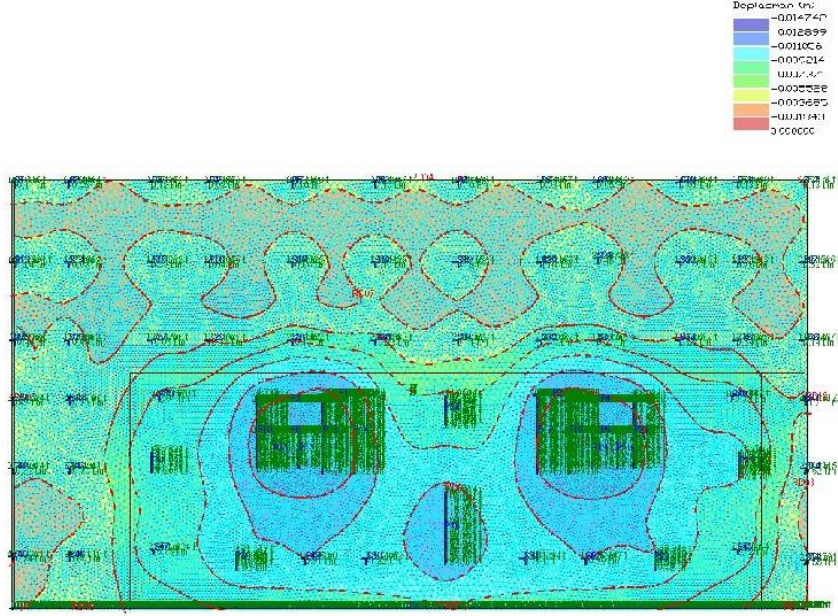
Çizelge 3.1: Sap 2000 Analiz Programı

TABLE: Frame Section Properties 01 - General

SectionName	Material	Shape	t3	t2	tf	tw	t2b	tfb	Area	TorsConst	I33	I22	I23	AS2	AS3	S33	S22	Z33	Z22	R33	R22	ConcCol	ConcBeam	Color	TotalWt	TotalMass	FromFile	A1Mod	A2Mod	A3Mod	J1Mod	I2Mod	I3Mod	M1Mod	W1Mod	SectionFile	Notes	
Text	Text	Text	m	m	m	m	m	m	m2	m4	m4	m4	m4	m2	m2	m3	m3	m3	m3	m	m	Yes/No	Yes/No	Text	KN	KN-s2/m	Yes/No	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless	Text	Text	
HE3600A	Steel	I/Wide Flange	0,35	0,35	0,0175	0,01	0,35	0,0175	0,11	1,033	0,0079	7,89	0,0000	0,0075	0,0008	0,00189085	0,0005258	0,0020	0,0008	0,0152	0,0074	N	N	Red	307.0	313.1	Y	1	1	1	1	1	1	1	1	1	HE3600A	Imported 16.06.2016 23:44:21 from EURO.PRO
HE3600M	Steel	I/Wide Flange	0,68	0,63	0,045	0,021	0,63	0,045	0,158	1,033	0,0069	7,89	0,0001	0,0040	0,0020	0,00843413	0,00124459	0,0000	0,0000	0,0274	0,0071	N	N	Blue	892.4	910.0	Y	1	1	1	1	1	1	1	1	1	HE3600M	Imported 16.06.2016 23:46:41 from EURO.PRO
IP2700	Steel	I/Wide Flange	0,27	0,31	0,0109	0,0065	0,0109	0,0065	0,100	1,059	0,0059	4,2E-05	0,0000	0,0017	0,0002	0,00042888	6,22E+09	0,0000	0,007E	0,112	0,030	N	N	Green	885.4	902.9	Y	1	1	1	1	1	1	1	1	1	IP2700	Imported 18.06.2019 03:25:42 from EURO.PRO
IP6000	Steel	I/Wide Flange	0,62	0,62	0,019	0,012	0,019	0,012	0,165	1,065	0,0092	3,39	0,0000	0,0072	0,0006	0,00396666	0,00030790	0,0000	0,0035	0,242	0,046	N	N	Yellow	864.5	881.6	Y	1	1	1	1	1	1	1	1	1	IP6000	Imported 16.06.2016 23:46:23 from EURO.PRO
Koris / 50	Concrete	Rectangular	0,6	0,6					0,36	0,01	0,008	0,008	0,003	0,003	0,036	0,036	0,036	0,04	0,04	0,173	0,173	N	Y	88	3.778	38.53	N	6	6	6	6	6	6	1	1		Added 16.06.2016 23:48:39	
Kolon / 50	Concrete	Rectangular	0,6	0,6					0,36	0,01	0,008	0,008	0,003	0,003	0,036	0,036	0,036	0,04	0,04	0,173	0,173	Y	N	16	0	0	N	6	6	6	6	6	6	1	1		Added 17.06.2016 00:26:52	

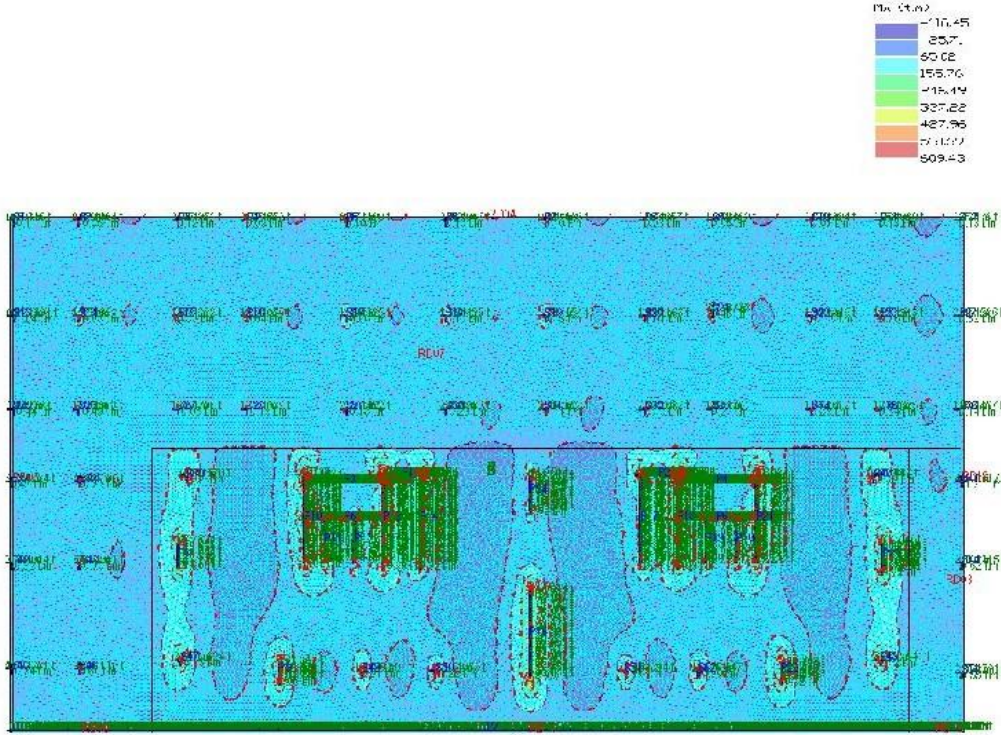
3.6 ProBina Tasarım Görüntüleri

Programın çalışma şekli ile alakalı 2.1. Tasarım Gereksinimleri ve Süreçleri başlığı altında açıklandığı gibi yapı paket program olup karma yada tek bütün çerçevede ki yapıların analizleri üzerinden tasarlanmış programdır.



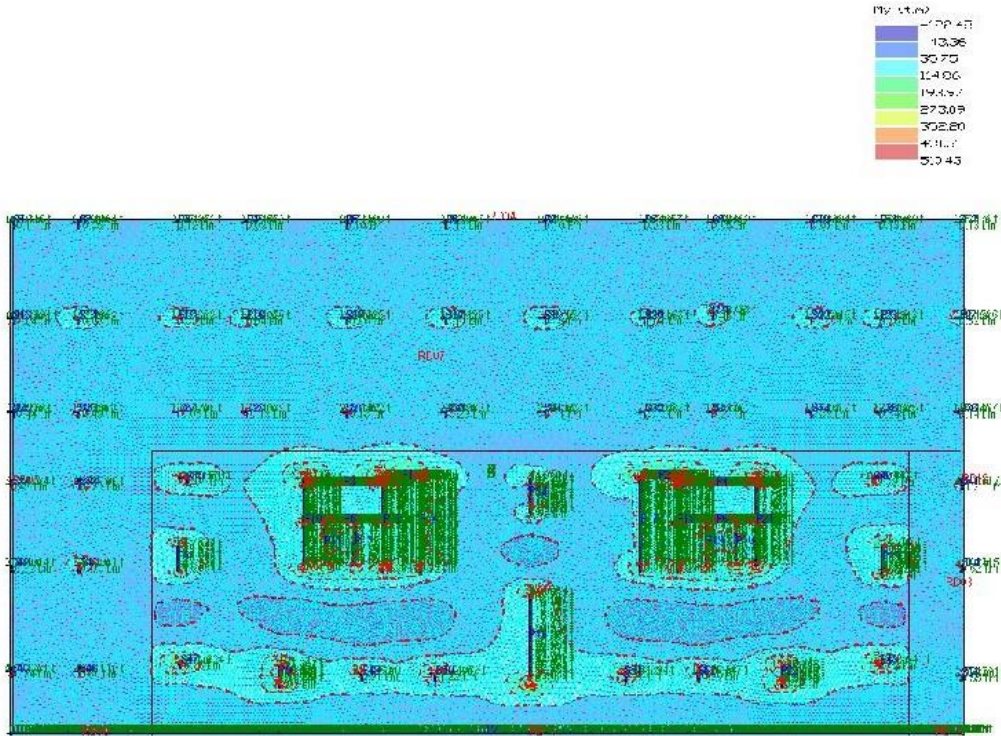
Şekil 3.8: ProBina Temel kodu deplasman dayanımı

C40 beton kullanılmış ve betonarme döşemesinin çelik ile güçlendirildiğinde zımbalama bölgeleri dahil zorlama olmadığını şekilden de anlaşılabilmektedir.

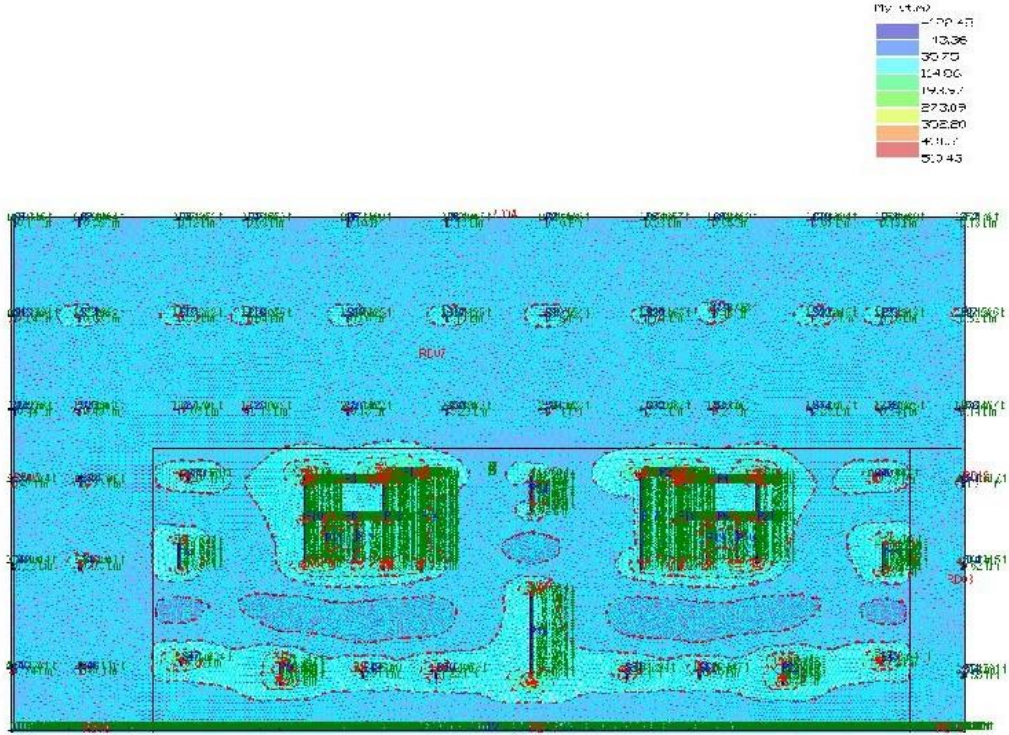


Şekil 3.9:ProBina Temel kodu moment-x yönü dayanımı

Döşeme zımpalama tehlikesi sınırında olan bölgelerde yükler yerinde sınırlı ve zorlama olmadığı döşeme analizinden anlaşılabilir.

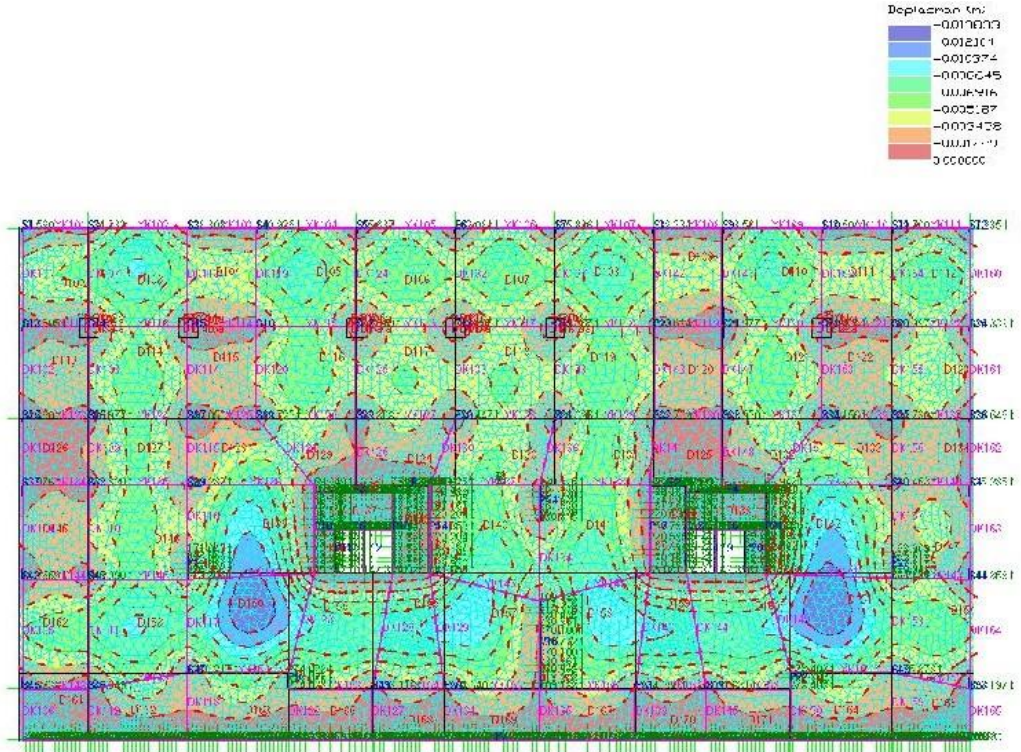


Şekil 3.10:ProBina Temel kodu moment-y yönü dayanımı

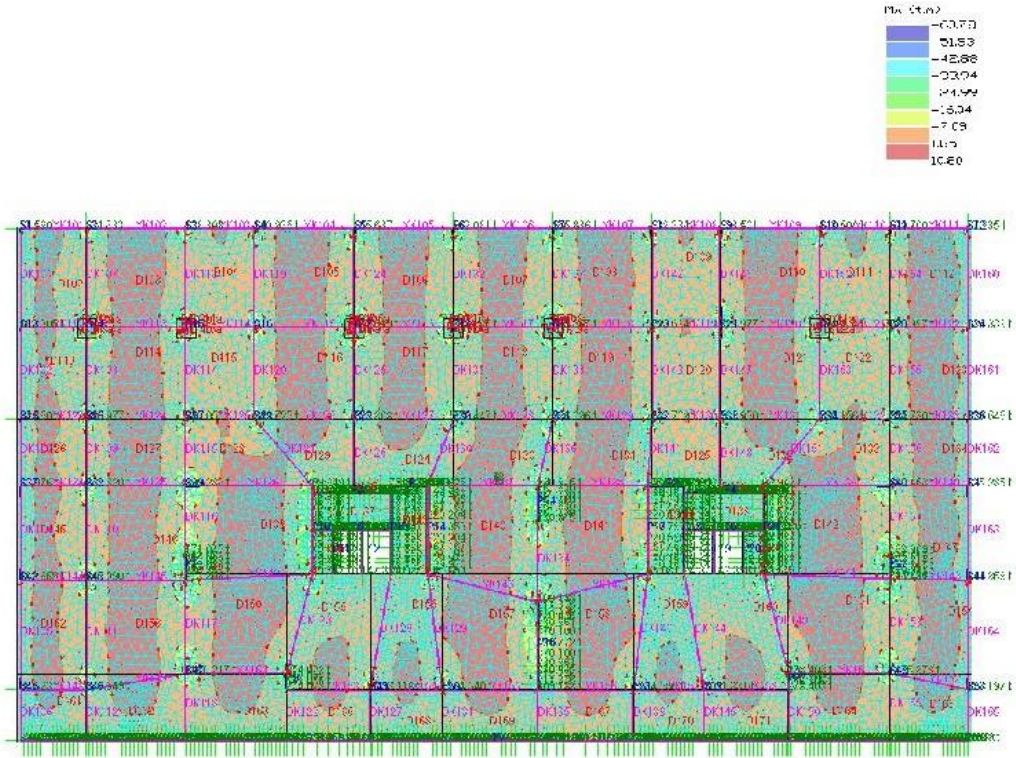


Şekil 3.11:ProBina Temel kodu moment-x ve moment-y yönü dayanımı

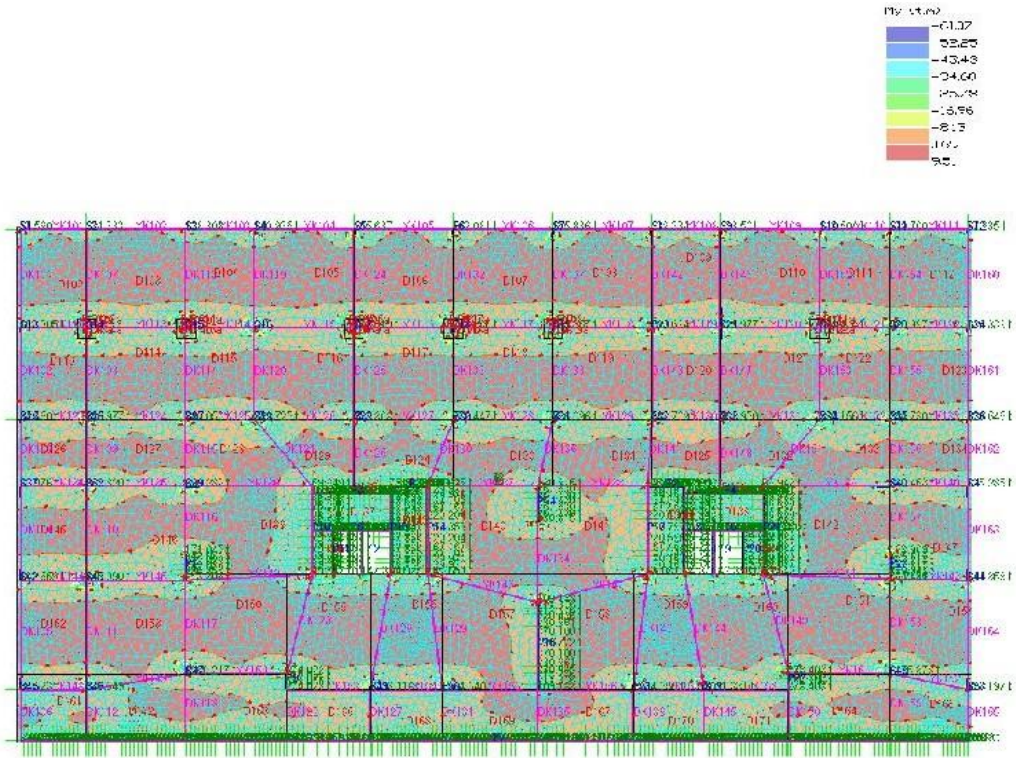
Döşemeye etki eden çift yönlü momentler sınır içerisinde olduğu anlaşılmaktadır. Bu da çelik ile güçlendirme sonucu mühendislik açısından güvenli bölgede kalma sınırını sağlamaktadır.



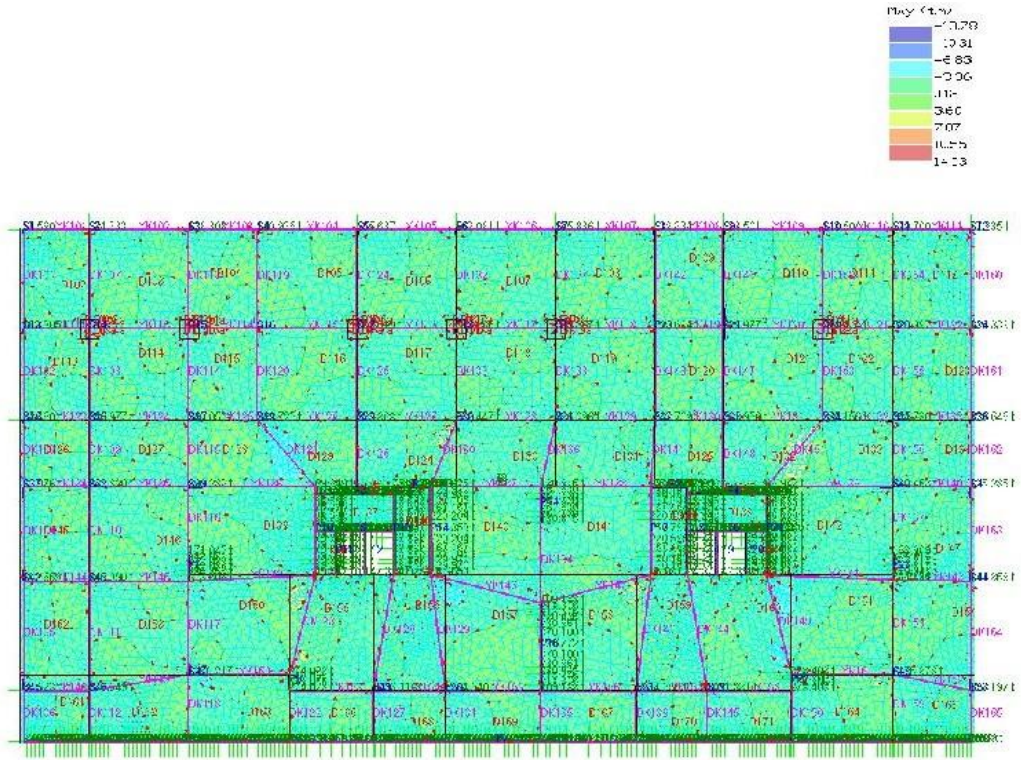
Şekil 3.12: ProBina 1.Kat kodu deplasman dayanımı



Şekil 3.13: ProBina 1.Kat kodu moment-x yönü dayanımı



Şekil 3.14: ProBina 1.Kat kodu moment-y yönü dayanımı



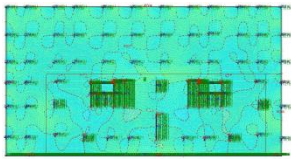
Şekil 3.15: ProBina 1.Kat kodu moment-x ve moment-y yönü dayanımı

Şekil 3.12, Şekil 3.13, Şekil 3.14 ve Şekil 3.15 de moment yüklemelerinde güvenli bölge sınırında kalınmış olup zımpalama gibi tehlike durumlar engellenmiştir.

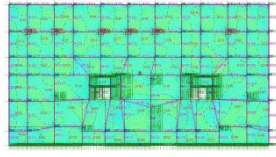
3.7 ProBina Yapı analizi

ProBina v18 kullanılarak elde edilmiş olan betonarme yapının merdiven hesabından örneklemeler aşağıda görülmektedir. Yapı döşeme alanında zorlanma yaşamazken birleşim alanlarında kesme kuvveti alanı üzerinden zorlamalar yaşanmıştır buna da önlem olarak bulon üzerinden geçiş yapılmıştır.

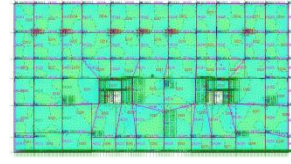
3.7.1 ProBina Döşeme Yük Analizi



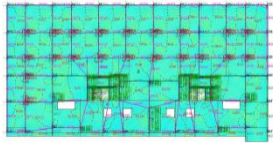
Temel Kodu Moment



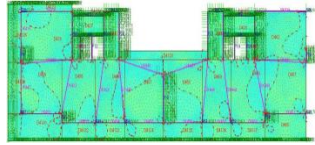
1.Kat Moment



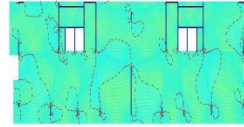
2.Kat Moment



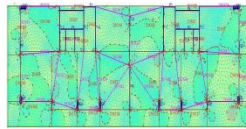
3.Kat Moment



4.Kat Moment



5.Kat Moment



6.Kat Moment

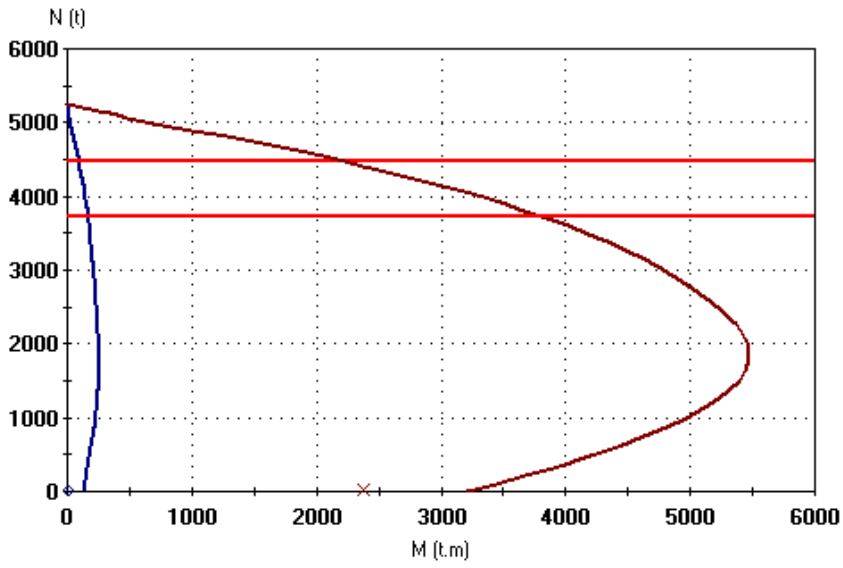
Şekil 3.16: ProBina Yapı kodlarının moment x-y göstergeleri

Yapıların tasarım planlaması ve düşüncesinde genel prensip limit durumları sınırları ile ayarlanmaktadır. Bu limit durumları planlamasında ve tasarımlarında 2 farklı metod vardır. Birinci metod taşıma limit durumu ki buna Hesap Kesit Mukavemet Dayanım durumu da söylenir. Bu hesaplamaların sonucunda kat sayılarla küçültülerek güvenli alanda kalma politikası uygulanmış olur. İkinci metod ise İşletme Limit Durumu yani Kuvvet Kesiti olarak söylenir. Bu hesaplamaların sonucunda da kat sayılarla büyütülür ve sonuç iki tasarım metodları arasında kat sayıların çarpanı kadar fark oluşmuş olur buda güvende kalma politikası gereğidir. Yapı döşemelerine baktığımızda işletme durumuna gelmediği halde yükleme sınırlarına yaklaşmamış olan yapının işletme yükleriyle beraber tasarım limitlerine yaklaşacağından tasarım ömrü boyunca sürekli yük yüklemelerinden dolayı gama ve yük katsayılarıyla güvenli bölgede kalınıyor. Güvende kalmak yüklerin kullanım ve tasarım ömrü boyunca yeterli olursa da yapının ani yük değişimleri yada yapı yüklemelere

karşı davranış şekli deęiřtikce yapı sistematięi yani dengesi bozulduęunu düşünürsek yapıyı taşıyıcı sistemler üzerinden sarıp kollayarak daha dayanıklı dayanım elde edilebiliyorken yapıyı güçlendirip rijitlik arttırırken süneklilięi azalacağından dolayı yapıyı daha stabil hale dönüřtürebilmek için mobilize montajlı ve güçlendirici taşıyıcı sistemlere geçilmesi gerekmektedir.

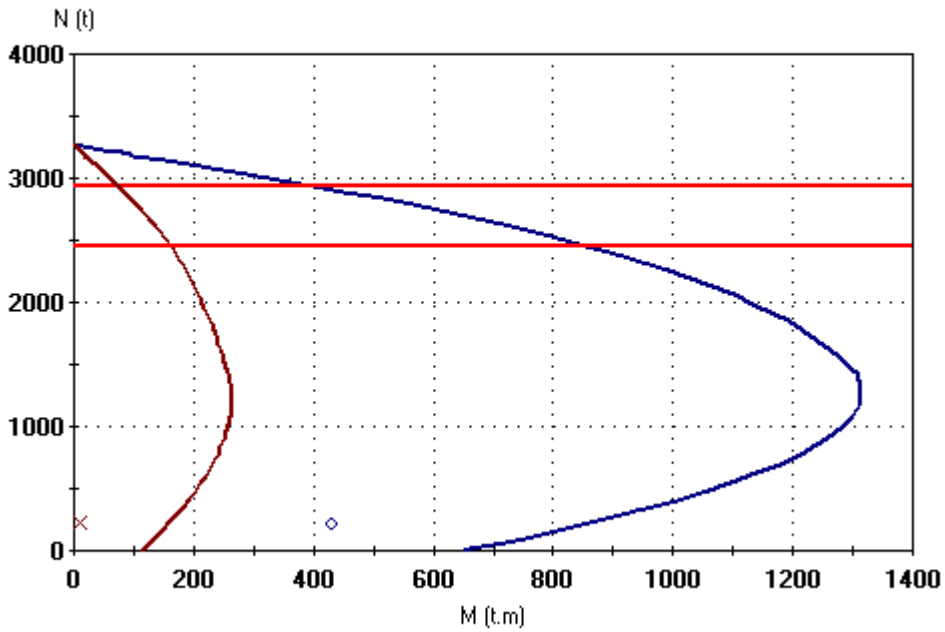
Yapıda görüldüęü gibi zorlama sınırları içerisinde olduęundan yapı yorulmaya başlamadıęından ve taşıyıcı sistemlerinde sorun olmadığından yapının da sürekli yenilenebilir olamayacağı düşünöldüęünde yapı güçlendirmenin yanında korunabilmeli yada yapı kenetlenmenin dışına çıkması engellenebilmeli bu durumda yapı daha stabil çalışabilecektir. Çelik ile güçlendirilmiş betonarme yapılarda yapı süneklilięi ve rijitlięi konusunda hiçbir fedakarlık gösterilmedięinden yapı kenetlenmesi konusunda da sorunsal bir durum olmadığından yapı daha stabil çalışabilmektedir.

3.7.2 Döşeme Yükleme Grafikleri



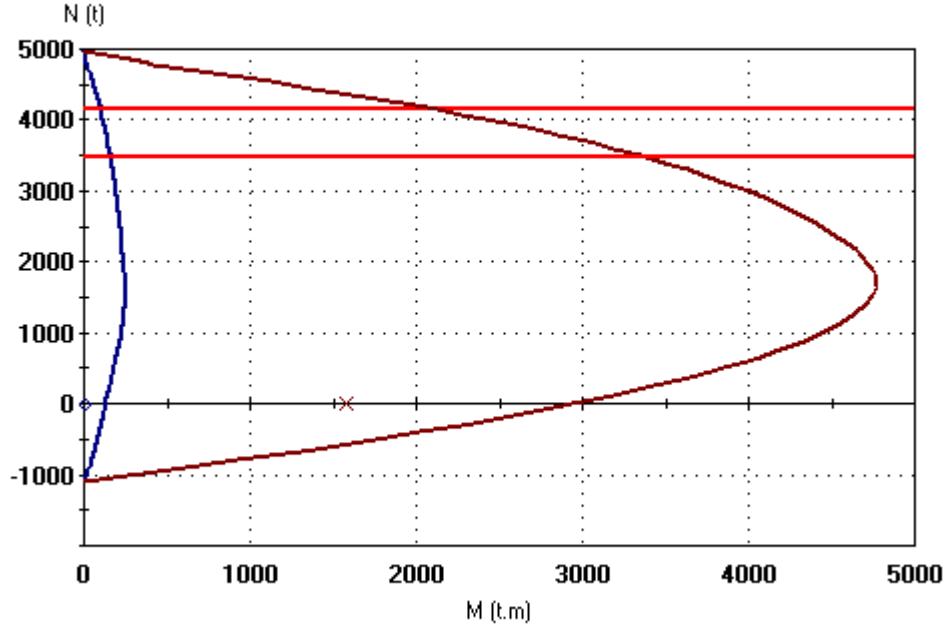
Şekil 3.17: ProBina Yüklemeler (Kombinasyon G+QF)

Moment grafiğinden anlaşıldığı gibi yapı normal yükleme içerisinde zorlama olmadığı anlaşılmaktadır.



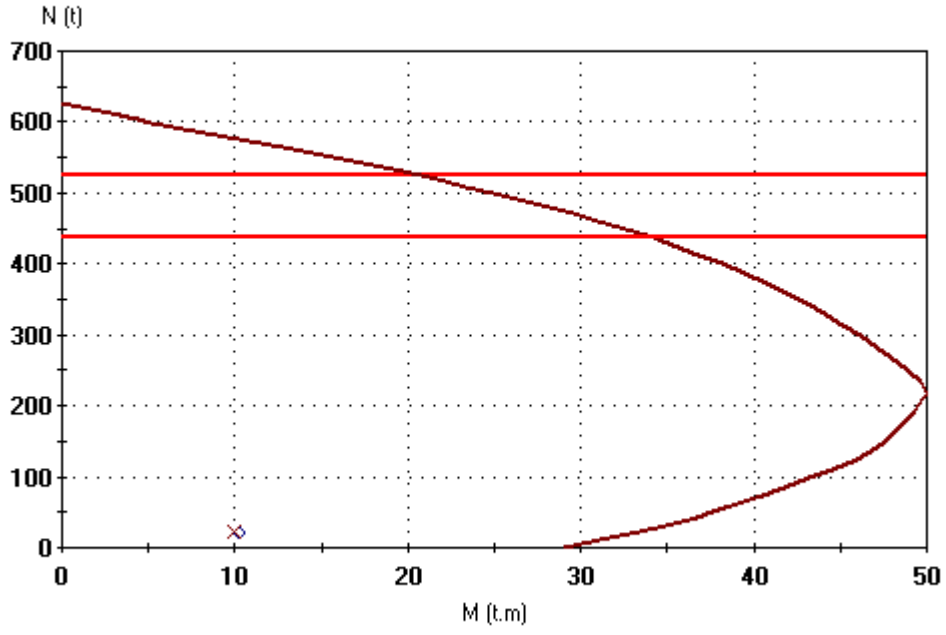
Şekil 3.18: ProBina Yüklemeler (Kombinasyon G+Sx+)

Moment grafiğinden anlaşıldığı gibi yapı normal yükleme içerisinde istenilen güvenli bölge içinde kalarak zorlama olmadığı anlaşılmaktadır.



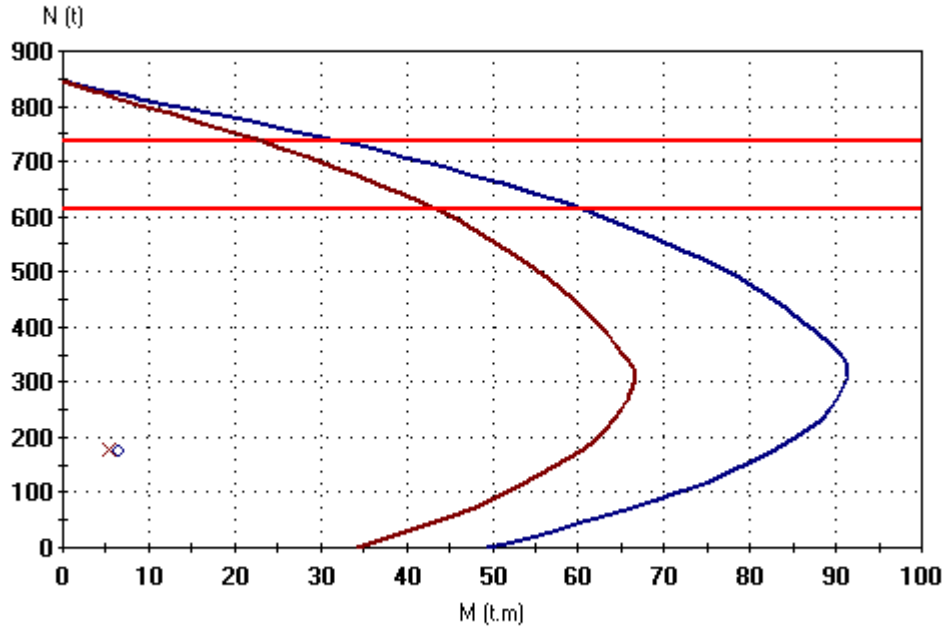
Şekil 3.19: ProBina Yüklemler (Kombinasyon G-Sx-)

Moment grafiğinden anlaşıldığı gibi yapı normal yükleme içerisinde yine yatay yükler dahil olarak ters tepkilemelerle zorlama olmadığı anlaşılmaktadır.



Şekil 3.20: ProBina Yüklemeler (Kombinasyon G-Sy+)

Moment grafiğinden anlaşıldığı gibi yapı normal yükleme içerisinde üst sınıra yaklaşmasına rağmen zorlama olmadığı anlaşılmaktadır.



Şekil 3.21: ProBina Yüklemeler (Kombinasyon G-Sy-)

Moment grafiğinden anlaşıldığı gibi yapı normal yüklemeye içerisinde yine yatay yükler dahil olarak ters tepkilemelerle zorlama olmadığı anlaşılmaktadır.

4. SONUÇ VE ÖNERİLER

Bu akademik çalışmada TBDY-2018 yönetmeliğine uygun olarak SAP2000, Seismo Struct ve ProBina kullanılmış olup Kocaeli depremi referans alınarak analizler yapılmıştır. Mühendislik açısından uygun hem de maddi manevi kayıplar önlemek açısından mümkün olan yapı inşası önemlidir. Yapının inşası süresi ve kullanılan malzemeler ve yapının kullanım ömrü boyunca yapılan bakım ve onarım maliyetleri ve yapılar düşünüldüğünde yapının maliyeti çelik-betonarme yapıdan daha fazla olmaktadır. Çelik ile güçlendirilmiş betonarme inşa edilmiş yapı rijitlikten taviz vermeden sünekliliği artırılabilen ve deprem gibi yapıyı ciddi zorlayan ve zarar verebilen doğal afetler düşünüldüğünde çelik malzemenin şuan ki teknolojilerle yangına karşı dayanımı kısmi artılabilmekte olup depreme karşı da dayanımı artırıldığında yapı deprem ve yangın gibi afetlere karşı daha güçlendirilebilir. Yapı tek bir dinamiğe sahip olduğunda bitişik nizam yapılara göre daha az etkilenmektedir. Bu akademik çalışma içerisinde aynı temel üzerinden inşa edilmiştir. Yapı 3 ayrı yapı gibi düşünülebilir. Bunu asansör ve kullanım alanları üzerinden tasarlanmıştır. Orta çekirdek haricindeki sağ ve sol çekirdekler çelik ile inşa edildiğinde yapının inşası süresi ile bakım ve onarım maliyetleri düşünüldüğünde ekonomik açıdan daha uygundur. Bu yüzden de tasarım anlamında ilk maliyet göz önüne alındığında daha uygun olan betonarme yapılara göre tasarım ve kullanım ömrü ile düşünüldüğünde ÇİGBY daha az bakıma ihtiyaç duyup daha fazla kullanım ömrü verebilmektedir. Yapının sünekliliğinden taviz verilmemesi gerektiğini söyleyen Türk yönetmeliğine göre bu konuda mühendislik ve tasarım açısından sınırlarla belirlenmiş çerçeveler içerisinde kalınması gerektiğini belirtmektedir. Eğer yapı rijitliği güvenlik sınırı içerisinde inşa edildikten sonra sünekliliği artırılabilirliğinden dolayı sismik izolatör gibi görev göreceğinden kullanım ömrü uzatılıp insan sağlığını da göz önünde bulundurduğunda yapı tasarım ömrü daha çok kullanıcı dostu, dönüşebilirliğe ve geleceğe dönük inşa edilmesi konusunda daha tercih edilebilmektedir.

Elde edilen verilere göre betonarme yapının dayanımı ile çelik ile güçlendirilmiş betonarme yapının dayanımı kıyaslamasından elde edilen veriler ışığında çelikle güçlendirilmiş yapıların hem daha az malzeme ile daha çok dayanım hem de inşa+bakım maliyeti daha uygun hem uygulanabilirliği ve değişikliği daha kolay ayrıca yapının kullanım amacına göre değişikliklerin sonucundaki güçlendirilmesi konusunda daha pratik bir yapı olduğu anlaşılmıştır. Yapıların kıyaslaması olarak düşünüldüğünde betonarme yapının kullanım ömrü boyunca ki bakım maliyeti değerlendirildiğinde çelikle güçlendirilmiş betonarme yapı %57 daha tasarruflu olduğu anlaşılmaktadır. Yapının inşa süresince kullanılan malzemeler, işçilik maliyeti ve geçen süreç değerlendirildiğinde çelikle güçlendirilmiş betonarme yapı daha kısa süreçte yapılabilir. Yapının yenileme gereksinimi olduğunda bakım ve onarım veya değişim ve imalat göz önüne alındığında çelikle güçlendirilmiş betonarme yapı insanların daha az mağdur etmektedir. Çünkü çelik malzemenin montajı ve imalatı betonarme imalatına göre 6 katlı yapıda yaklaşık %20 daha hızlı yapılabilir. Ayrıca onarım ihtiyacı olduğunda betonarme yapılarda yerel çözüm yapılamazken çelikle güçlendirilmiş betonarme yapılarda bu daha uygulanabilir. Türkiye deki yakın zamandaki en büyük deprem olan Kocaeli depremi baz alındığında çelikle güçlendirilmiş betonarme yapı dinamik olarak depremin pick anlarını ve depremin tamamını az deplasman yaparak geçiren yapı ayrıca daha az yorulduğundan kullanım ömrü yada kullanım alanı azalmamıştır. Kısacası bu veriler ve analizler eşliğinde anlaşıldığı gibi çelikle güçlendirilmiş betonarme yapı kullanım pratikliği ve kullanım ömrü ve dayanımı ve inşası ile bakımı düşünüldüğünde daha uygun ve uygulanabilir olmaktadır. Ayrıca kullanım ömrü esnasında yapının tasarım amacı dışında kullanımlardan kaynaklanan yapının sehimi konusunda daha dayanımlı olduğundan yapı daha uzun ömürlü yada ömrünün tamamını daha güvenli kullanımına mümkündür. Sonuç olarak yapı ilk inşa esnasındaki kullanılan malzeme maliyeti hariç her aşamada hem daha dayanımlı ve hem daha uygun hem daha kullanılabilir. Ayrıca yapının sadece inşası olarak düşünülmemeli ve bu kullanım esnasında yada yapımı aşamasında hepsi bir bütün olarak düşünmek gerekmektedir. Buna göre de çelikle güçlendirilmiş betonarme yapı kullanıma göre daha uygundur. Doğal afetler düşünüldüğünde, yangına karşı olarak betonarme yapıya göre büyük ayrıcalıklı performansı var denemez ama

deprem düşünülduğünde yapı hafif ve dinamiğe uygun olduğundan daha labil ve daha dayanıklı olduğu yapılan analiz sonuçlarından elde edilebilmektedir.

Yapı güçlendirmesinde yapının çerçevesinde ana iskelet betonarme olduktan sonra yapının temelden aldığı güçte betonarme olduğundan basınca karşı dayanım özelliğiyle yapının ilk etkiden sonra yapıyı deprem aynı frekansta hareket ettirebilmek için yapıyı temel üstü sağ ve sol çerçeveleri çelik ile güçlendirildiğinde yapı hem ağırlığından dolayı ağırlık merkezi hem yapının çekme dayanımı hem frekansı deprem uyumlu olduğundan yapı daha az zarar görmüş oluyor ayrıca yapı onarımı da daha uygulanabilir olduğundan dolayı yapı onarımı ve bakımı daha pratik ve mümkün olmaktadır.

Bu çalışmanın içerisinde, 1999 yılında olmuş olan Kocaeli depremi en yakın merkez olan ve İstanbul'da ki sismografi tarafından ölçülen değerler referans alınmıştır. Bu değerlere göre yapının orta çekirdeğinin betonarme olmasına rağmen sağ ve sol çekirdeklerinin sünekliliğinden kaynaklı olduğu birleşim noktalarındaki etkilemelerinden anlaşıldığına göre yapı sünekliliğini artırmış olmakta aynı anda rijitlik açısından da rijitliğini korumakta olduğundan gerilmeyi de azalttığı anlaşılmakta olup yapının kullanım ömrü boyunca yapı doğal afetlere karşı daha stabil dayanım gösterebilmektedir. Yapı güçlendirmesi esnasında çekirdek çekirdek bölüm oluşturmanın yanı sıra yapıyı temel üstünde çelik inşa olarak yapılmış çalışmaların daha güçlendirmeye ihtiyaç duyulduğu akademik çalışmalarla kanıtlanmış durumdadır. Yapıyı temel üstünde bütün karkası çelik olduğunda deprem afetine karşı dayanımı yetersiz olduğu görülmektedir. Orta çekirdek betonarme olduğunda dayanımın yüksek olduğunu bu çalışma içerisinde görülebilmektedir. Yapıyı temel üstünde enlem ve boylam oranında bir formül yada stabilitesini koruduğu sürece minimum betonarme maksimum çelik inşa edilebileceği ile alakalı çalışma ile bu çalışma geliştirilebilir.

5. KAYNAKÇA

- A. Molinari, M. Pellizzari, G. Straffelini, and M. Pirovano, "Corrosion behaviour of a surface-treated AISI H11 hot work tool steel in molten aluminium alloy," *Surface and Coatings Technology*, vol. 126, no. 1, pp. 31–38, 2000.
- Kılıç A. (2008).** Çelik Taşıyıcıların Yangın Yalıtımı. *Yangın ve Güvenlik dergisi* sayı:118, s.8-12.
- Akbaş B. (2011).** "Endüstriyel çelik yapıların sismik detaylandırılması", 1. Türkiye Deprem Mühendisliği ve Sismoloji Konferansı, (2011)
- ALTUN A., YILMAZ F., YILMAZ C. (2006).** " Çelik Lif Katkılı ve Katkısız Betonarme Kirişlerin Basit Eğilme ve Patlama Yükleme İle Davranışlarının İncelenmesi" *Erciyes Üniversitesi FBE Dergisi* 22(1-2) 112-120, 2006.
- Avcı, Y., ve U. Yazgan (2018).** "Zemin-yapı etkileşimli sistemlerde en büyük doğrusal olmayan yer değiştirme oranı", *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 18(2):2-17.
- C.H.Gur (2017).** Kaynak Dikişlerinin Tahribatsız Yöntemlerle Muayenesinde Avrupa Standardları. *Kaynak Teknolojisi II. Ulusal Kongresi*, 231-237
- Celikag M. (2002).** and Bozkir, S., "Feasibility of Using Structural Steel in North Cyprus and Turkey", 7th International Conference on Steel and Space Structures, 2-3 October 2002, Singapore, pp.373-380.
- Celikag, M., and Ozbilen, M.,** "Inadequate Applications of Construction Industry in North Cyprus and Recommendations", 11th International Conference on Inspection Appraisal, Repair and Maintenance of Structures, 14-17 November 2007, North Cyprus.
- Özhendekci D. (2017).** Çelik Yapılar I Ders Notları YTÜ İnşaat Müh. Böl., İstanbul.
- Darılmaz K. (2007),** SAP 2000 ile Yapı Sistemlerinin Analizi, TMMOB İnşaat Mühendisleri Odası İzmir Şubesi Seminer Notları, Şubat 2007
- Düzel E, (2010).** "Düşeyde rijitlik düzensizliği bulunan çerçevelerin sönüm elemanları ve çelik çaprazlar ile rehabilitasyonu" *Yüksek lisans tezi*, Niğde Üniversitesi, (2010)
- Erten, E., Taskin, G., Lopez-Sanchez, JM.** "Selection of PolSAR Observables for Crop Biophysical Variable Estimation With Global Sensitivity Analysis", *IEEE Geoscience and Remote Sensing Letters*, 2019, DOI:10.1109/LGRS.2019.2891953
- European Convention For Constructional Steelwork. (1993).** Examplesto EC3. (First Edition). Brussel.
- F. Demirel, E. Özkan. (2003).** Çelik Yapı Bileşenleri ve Yangın Güvenlik Önlemleri. *Gazi Üniv. Müh. Mim. Fak. Der. Cilt 18, No 4*, 89-107, 2003.

- F.Pirođlu, E. Uzgider, M. Vural, Ö.B. Çađlayan (2003).** Geçmişten Bugüne Yapı Çeliđi ve Önemli Yapısal Özellikleri. TMMOB, Türkiye Mühendislik Haberleri Dergisi, 426 - 2003/4, 43-46.
- Fardis, M., Carvalho, E., Elnashai, A., Faccioli, E., Pinto, P. ve Plumier., A. (2005).** Designers' Guide To EN 1998-1 and EN 1998-5 Eurocode 8: Design Of Structures For Earthquake Resistance London: Thomas Telford Ltd.
- G.Berkmen (2017).** Endüstriyel Yapılarda Yangın Yalıtım Uygulamaları TMMOB Makina Mühendisleri Odası Yalıtım Kongresi
- G.Gedge (2008).** Structural uses of stainless steel — buildings and civil engineering. Journal of Constructional Steel Research 64 (2008) 1194–1198
- H. Arık (2016).** Demir Çelik Metalürjisi Ders Notları, Gazi Üniversitesi, Ankara
- I. Schruuff, "Te tool steel producer's contribution to successful die casting of structural components," in Proceedings of the European Conference on High Tech Die Casting (HTDC '12), Vicenza, Italy, 2012.
- I.Vayas, P. Thanopoulos. Innovative Dissipative (INERD) Pin Connections for Seismic Resistant Braced Frames. Steel Structures, 5, 2005, 453-463.
- İ.Özkan (2011).** AISI304 Östenitik Paslanmaz Çeliklerin Çeşitli Kaynak Uygulamaları, Kocaeli Üniversitesi, Müh.Fak., Bitirme Projesi, 2011
- İstanbul Teknik Üniversitesi Deprem Mühendisliđi ve Afet Yönetimi Enstitüsü,2017**
- K.-D. Fuchs, "Hot-work tool steels with improved properties for die casting applications," in Proceedings of the 6th International Tooling Conference, Karlstad University, Karlstad, Sweden, 2002.
- K.Yemez (2015).** Çelik Yapı Tasarımını Etkileyen Son Dönemlerdeki Araştırma ve Gelimseler. TMMOB İnaaat Mühendisleri Odası 6. Ulusal Çelik Yapılar Sempozyumu
- KHOURY, G.A. (1992),** "Compressive Strength of Concrete at High Temperatures: A Reassessment", Magazine of Concrete Research, 44 (161), 291-309, 1992.
- Kıymaz, G. (2010),** "Çelik Yapıların Depreme Karşı Güçlendirilmesi", Dünya İnşaat Dergisi,(2010)
- Kirchhoff G. Ueber** das Gleichgewicht und die Bewegung eines unendlich dünnen elastischen Stabes. J Für Die Reine Und Angew Math, p 285–313. <http://dx.doi.org/10.1515/crll.1859.56.285>.
- Koiter WT.** A consistent first approximations in the generaltheory of thin elastic shells. In: Proc Symp. Theory Thin Elastic Shells, 1959, p. 12–23.
- Küçük, Ç., Taşkın, G., Erten, E. (2016)** Paddy-Rice Phenology Classification Based on Machine-Learning Methods Using Multitemporal Co-Polar X-Band SAR Images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 9, No. 6, 06/2016, pp. 2509 - 2519.
- Michael D.Symans. Seismo Struct Systems: Seismic Protection
- P. Gumpel, T. Bogatzky, A. Huber, and B. Geigges (2002),** "Comparison of different characteristics of modern hot-work tool steels," in Proceedings of the 6th International Tooling Conference, 2002.

- S.Kurt,C.O.Azelođlu** Asansörlerde Kullanılan Çelik Tel Halatlar, Seçim ve Bakım Yöntemleri, www.mmo.org.tr(27-12-2018)
- Naimi S. (2016)** Çelik Yapıların Ders Notları, İstanbul Aydın Üniversitesi, İstanbul, 2016.
- Naimi S. (2017)**, Çelik Yapıların Tasarımı Ders Notları, İstanbul Aydın Üniversitesi, İstanbul, 2017.
- Tama,Y. (2010)**, "Çelik Yapılarda Bayrak Levhalarının Sismik Davranışlarının Araştırılması", Osmangazi Üniversitesi Müh.Mim.Fak. Dergisi. (2010)
- Taşkın, G., Ersoy, O.K., Kamaşak, M.E., (2015)** Earthquake-induced damage classification from postearthquake satellite image using spectral and spatial features with support vector selection and adaptation. Journal of Applied Remote Sensing, Vol. 9, No. 1, 08/2015, pp. 096017/1-20.
- TS 500 Turkish Standards "Requirements For Design And Construction Of Reinforced Concrete Structures", February 2000. Ankara, Turkey
- TS648, Çelik Yapıların Hesap ve Yapım Kuralları, Aralık 1980
- Türkiye Bina Deprem Yönetmeliđi 2018
- W. Ecker, T. Antretter, and R. Ebner (20017)**, "Calibration and validation of an elasto-viscoplastic material model for a hot work tool steel used in pressure casting dies," in Proceedings of the 10th International Conference on Mechanical Behavior of Materialslocation, S. W. Nam, Y. W. Chang, S. B. Lee et al., Eds., pp. 27–31, Busan, Republic of Korea, 2007.
- Yazgan, U., R. Oyguc, M.E. Ergüven ve Z. Celep, (2016)**. "Seismic performance of buildings during 2011 Van earthquakes and rebuilding efforts", Earthquake Engineering and Engineering Vibration 15(3):591-606
- Yurdakul A., Pirođlu F., Okay N. (2013)** "Kocaeli Derince Eğitim ve Araştırma Hastanesi Mevcut Afet Planı Çerçevesinde Çalışanların Afete Hazırlılıđının Deđerlendirilmesi", Mehmet Akif Ersoy Üniversitesi, Sađlık Bilimleri Enstitüsü Dergisi, Cilt 1, Sayı 1, Sayfa 75-85, 2013.

EKLER

Ek 1: Seismo Struct Analiz Raporu Tamamı

Ek 2: ProBina Analiz Raporu Tamamı

Ek 3: Sap2000 Analiz Raporu Tamamı

Ek 1: Seismo Struct Analiz Raporu Tamamı

Analysis completed on

Date: 07.06.2019

Time: 00:03:43

SeismoStruct v2016 build 1

Estimated size of output file: 204261 KB

Permanent Loading

LF= 1,00000, LF_incr= 1,00000 (Iter: 2 => Converg)

Variable Loading

Time

Time= 0,01000, dt= 0,01000 (Iter: 1 => Converg)

Time= 0,02000, dt= 0,01000 (Iter: 1 => Converg)

Time= 0,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,28000, dt= 0,01000 (Iter: 1 => Converg)

Time= 0,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,54000, dt= 0,01000 (Iter: 1 => Converg)

Time= 0,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,80000, dt= 0,01000 (Iter: 1 => Converg)

Time= 0,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 0,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,06000, dt= 0,01000 (Iter: 1 => Converg)

Time= 1,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,32000, dt= 0,01000 (Iter: 1 => Converg)

Time= 1,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,58000, dt= 0,01000 (Iter: 1 => Converg)

Time= 1,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,84000, dt= 0,01000 (Iter: 1 => Converg)

Time= 1,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 1,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,10000, dt= 0,01000 (Iter: 1 => Converg)

Time= 2,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,36000, dt= 0,01000 (Iter: 1 => Converg)

Time= 2,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,62000, dt= 0,01000 (Iter: 1 => Converg)

Time= 2,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,88000, dt= 0,01000 (Iter: 1 => Converg)

Time= 2,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 2,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,14000, dt= 0,01000 (Iter: 1 => Converg)

Time= 3,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,40000, dt= 0,01000 (Iter: 1 => Converg)

Time= 3,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,66000, dt= 0,01000 (Iter: 1 => Converg)

Time= 3,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,92000, dt= 0,01000 (Iter: 1 => Converg)

Time= 3,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 3,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,18000, dt= 0,01000 (Iter: 1 => Converg)

Time= 4,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,44000, dt= 0,01000 (Iter: 1 => Converg)

Time= 4,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,70000, dt= 0,01000 (Iter: 1 => Converg)

Time= 4,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,96000, dt= 0,01000 (Iter: 1 => Converg)

Time= 4,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 4,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,22000, dt= 0,01000 (Iter: 1 => Converg)

Time= 5,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,48000, dt= 0,01000 (Iter: 1 => Converg)

Time= 5,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,74000, dt= 0,01000 (Iter: 1 => Converg)

Time= 5,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 5,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,00000, dt= 0,01000 (Iter: 1 => Converg)

Time= 6,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,26000, dt= 0,01000 (Iter: 1 => Converg)

Time= 6,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,52000, dt= 0,01000 (Iter: 1 => Converg)

Time= 6,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,78000, dt= 0,01000 (Iter: 1 => Converg)

Time= 6,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 6,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,04000, dt= 0,01000 (Iter: 1 => Converg)

Time= 7,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,30000, dt= 0,01000 (Iter: 1 => Converg)

Time= 7,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,56000, dt= 0,01000 (Iter: 1 => Converg)

Time= 7,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,82000, dt= 0,01000 (Iter: 1 => Converg)

Time= 7,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 7,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,08000, dt= 0,01000 (Iter: 1 => Converg)

Time= 8,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,34000, dt= 0,01000 (Iter: 1 => Converg)

Time= 8,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,60000, dt= 0,01000 (Iter: 1 => Converg)

Time= 8,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,86000, dt= 0,01000 (Iter: 1 => Converg)

Time= 8,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 8,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,12000, dt= 0,01000 (Iter: 1 => Converg)

Time= 9,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,38000, dt= 0,01000 (Iter: 1 => Converg)

Time= 9,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,64000, dt= 0,01000 (Iter: 1 => Converg)

Time= 9,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,90000, dt= 0,01000 (Iter: 1 => Converg)

Time= 9,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 9,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,16000, dt= 0,01000 (Iter: 1 => Converg)

Time= 10,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,28000, dt= 0,01000 (Iter: 2 => Converg)
Time= 10,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,32000, dt= 0,01000 (Iter: 2 => Converg)
Time= 10,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,42000, dt= 0,01000 (Iter: 1 => Converg)

Time= 10,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,68000, dt= 0,01000 (Iter: 1 => Converg)

Time= 10,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,94000, dt= 0,01000 (Iter: 1 => Converg)

Time= 10,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 10,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,20000, dt= 0,01000 (Iter: 1 => Converg)

Time= 11,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,46000, dt= 0,01000 (Iter: 1 => Converg)

Time= 11,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,72000, dt= 0,01000 (Iter: 1 => Converg)

Time= 11,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 11,98000, dt= 0,01000 (Iter: 1 => Converg)

Time= 11,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,24000, dt= 0,01000 (Iter: 1 => Converg)

Time= 12,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,41000, dt= (Iter: 1 => fbd_Ite)
Time= 12,41125, dt= 0,00125 (Iter: 1 => Converg)
Time= 12,41375, dt= 0,00250 (Iter: 1 => Converg)
Time= 12,41875, dt= 0,00500 (Iter: 1 => Converg)
Time= 12,42000, dt= 0,00125 (Iter: 1 => Converg)
Time= 12,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,46000, dt= 0,01000 (Iter: 1 => Converg)

Time= 12,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,53000, dt= (Iter: 0 => fbd_Ite)
Time= 12,53125, dt= 0,00125 (Iter: 1 => Converg)
Time= 12,53375, dt= 0,00250 (Iter: 1 => Converg)
Time= 12,53375, dt= (Iter: 0 => fbd_Ite)
Time= 12,53437, dt= 0,00062 (Iter: 1 => Converg)
Time= 12,53562, dt= 0,00125 (Iter: 1 => Converg)
Time= 12,53812, dt= 0,00250 (Iter: 1 => Converg)
Time= 12,54000, dt= 0,00188 (Iter: 1 => Converg)
Time= 12,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,56000, dt= (Iter: 1 => fbd_Ite)
Time= 12,56125, dt= 0,00125 (Iter: 1 => Converg)
Time= 12,56375, dt= 0,00250 (Iter: 1 => Converg)
Time= 12,56875, dt= 0,00500 (Iter: 1 => Converg)
Time= 12,57000, dt= 0,00125 (Iter: 1 => Converg)
Time= 12,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,61000, dt= 0,01000 (Iter: 1 => Converg)

Time= 12,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,68000, dt= 0,01000 (Iter: 2 => Converg)
Time= 12,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,87000, dt= 0,01000 (Iter: 1 => Converg)

Time= 12,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 12,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,13000, dt= 0,01000 (Iter: 1 => Converg)

Time= 13,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,39000, dt= 0,01000 (Iter: 1 => Converg)

Time= 13,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,65000, dt= 0,01000 (Iter: 1 => Converg)

Time= 13,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,90000, dt= (Iter: 1 => fbd_Ite)

Time= 13,90125, dt= 0,00125 (Iter: 1 => Converg)
Time= 13,90375, dt= 0,00250 (Iter: 1 => Converg)
Time= 13,90875, dt= 0,00500 (Iter: 1 => Converg)
Time= 13,91000, dt= 0,00125 (Iter: 1 => Converg)
Time= 13,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 13,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,13000, dt= 0,01000 (Iter: 1 => Converg)

Time= 14,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,39000, dt= 0,01000 (Iter: 1 => Converg)

Time= 14,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,64000, dt= 0,01000 (Iter: 2 => Converg)
Time= 14,65000, dt= 0,01000 (Iter: 1 => Converg)

Time= 14,66000, dt= 0,01000 (Iter: 2 => Converg)
Time= 14,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,68000, dt= 0,01000 (Iter: 2 => Converg)
Time= 14,69000, dt= 0,01000 (Iter: 2 => Converg)
Time= 14,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,91000, dt= 0,01000 (Iter: 1 => Converg)

Time= 14,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 14,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,17000, dt= 0,01000 (Iter: 1 => Converg)

Time= 15,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,43000, dt= 0,01000 (Iter: 1 => Converg)

Time= 15,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,69000, dt= 0,01000 (Iter: 1 => Converg)

Time= 15,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,95000, dt= 0,01000 (Iter: 1 => Converg)

Time= 15,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 15,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,21000, dt= 0,01000 (Iter: 1 => Converg)

Time= 16,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,47000, dt= 0,01000 (Iter: 1 => Converg)

Time= 16,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,73000, dt= 0,01000 (Iter: 1 => Converg)

Time= 16,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 16,99000, dt= 0,01000 (Iter: 1 => Converg)

Time= 17,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,25000, dt= 0,01000 (Iter: 1 => Converg)

Time= 17,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,51000, dt= 0,01000 (Iter: 1 => Converg)

Time= 17,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,77000, dt= 0,01000 (Iter: 1 => Converg)

Time= 17,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 17,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,03000, dt= 0,01000 (Iter: 1 => Converg)

Time= 18,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,29000, dt= 0,01000 (Iter: 1 => Converg)

Time= 18,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,55000, dt= 0,01000 (Iter: 1 => Converg)

Time= 18,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,81000, dt= 0,01000 (Iter: 1 => Converg)

Time= 18,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 18,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,07000, dt= 0,01000 (Iter: 1 => Converg)

Time= 19,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,33000, dt= 0,01000 (Iter: 1 => Converg)

Time= 19,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,59000, dt= 0,01000 (Iter: 1 => Converg)

Time= 19,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,85000, dt= 0,01000 (Iter: 1 => Converg)

Time= 19,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 19,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,11000, dt= 0,01000 (Iter: 1 => Converg)

Time= 20,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,37000, dt= 0,01000 (Iter: 1 => Converg)

Time= 20,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,63000, dt= 0,01000 (Iter: 1 => Converg)

Time= 20,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,89000, dt= 0,01000 (Iter: 1 => Converg)

Time= 20,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 20,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,15000, dt= 0,01000 (Iter: 1 => Converg)

Time= 21,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,41000, dt= 0,01000 (Iter: 1 => Converg)

Time= 21,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,67000, dt= 0,01000 (Iter: 1 => Converg)

Time= 21,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,93000, dt= 0,01000 (Iter: 1 => Converg)

Time= 21,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 21,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,19000, dt= 0,01000 (Iter: 1 => Converg)

Time= 22,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,45000, dt= 0,01000 (Iter: 1 => Converg)

Time= 22,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,71000, dt= 0,01000 (Iter: 1 => Converg)

Time= 22,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,97000, dt= 0,01000 (Iter: 1 => Converg)

Time= 22,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 22,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,23000, dt= 0,01000 (Iter: 1 => Converg)

Time= 23,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,49000, dt= 0,01000 (Iter: 1 => Converg)

Time= 23,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,75000, dt= 0,01000 (Iter: 1 => Converg)

Time= 23,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 23,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,01000, dt= 0,01000 (Iter: 1 => Converg)

Time= 24,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,27000, dt= 0,01000 (Iter: 1 => Converg)

Time= 24,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,53000, dt= 0,01000 (Iter: 1 => Converg)

Time= 24,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,79000, dt= 0,01000 (Iter: 1 => Converg)

Time= 24,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 24,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,05000, dt= 0,01000 (Iter: 1 => Converg)

Time= 25,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,31000, dt= 0,01000 (Iter: 1 => Converg)

Time= 25,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,57000, dt= 0,01000 (Iter: 1 => Converg)

Time= 25,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,83000, dt= 0,01000 (Iter: 1 => Converg)

Time= 25,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 25,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,09000, dt= 0,01000 (Iter: 1 => Converg)

Time= 26,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,35000, dt= 0,01000 (Iter: 1 => Converg)

Time= 26,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,61000, dt= 0,01000 (Iter: 1 => Converg)

Time= 26,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,87000, dt= 0,01000 (Iter: 1 => Converg)

Time= 26,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 26,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,13000, dt= 0,01000 (Iter: 1 => Converg)

Time= 27,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,39000, dt= 0,01000 (Iter: 1 => Converg)

Time= 27,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,65000, dt= 0,01000 (Iter: 1 => Converg)

Time= 27,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,91000, dt= 0,01000 (Iter: 1 => Converg)

Time= 27,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 27,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,17000, dt= 0,01000 (Iter: 1 => Converg)

Time= 28,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,43000, dt= 0,01000 (Iter: 1 => Converg)

Time= 28,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,69000, dt= 0,01000 (Iter: 1 => Converg)

Time= 28,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,95000, dt= 0,01000 (Iter: 1 => Converg)

Time= 28,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 28,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,21000, dt= 0,01000 (Iter: 1 => Converg)

Time= 29,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,47000, dt= 0,01000 (Iter: 1 => Converg)

Time= 29,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,73000, dt= 0,01000 (Iter: 1 => Converg)

Time= 29,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 29,99000, dt= 0,01000 (Iter: 1 => Converg)

Time= 30,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,25000, dt= 0,01000 (Iter: 1 => Converg)

Time= 30,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,51000, dt= 0,01000 (Iter: 1 => Converg)

Time= 30,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,77000, dt= 0,01000 (Iter: 1 => Converg)

Time= 30,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 30,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,03000, dt= 0,01000 (Iter: 1 => Converg)

Time= 31,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,29000, dt= 0,01000 (Iter: 1 => Converg)

Time= 31,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,55000, dt= 0,01000 (Iter: 1 => Converg)

Time= 31,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,81000, dt= 0,01000 (Iter: 1 => Converg)

Time= 31,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 31,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,07000, dt= 0,01000 (Iter: 1 => Converg)

Time= 32,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,33000, dt= 0,01000 (Iter: 1 => Converg)

Time= 32,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,59000, dt= 0,01000 (Iter: 1 => Converg)

Time= 32,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,85000, dt= 0,01000 (Iter: 1 => Converg)

Time= 32,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 32,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,11000, dt= 0,01000 (Iter: 1 => Converg)

Time= 33,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,15000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,37000, dt= 0,01000 (Iter: 1 => Converg)

Time= 33,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,41000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,63000, dt= 0,01000 (Iter: 1 => Converg)

Time= 33,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,67000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,89000, dt= 0,01000 (Iter: 1 => Converg)

Time= 33,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,93000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,94000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,95000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,96000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,97000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,98000, dt= 0,01000 (Iter: 1 => Converg)
Time= 33,99000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,00000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,01000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,02000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,03000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,04000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,05000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,06000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,07000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,08000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,09000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,10000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,11000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,12000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,13000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,14000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,15000, dt= 0,01000 (Iter: 1 => Converg)

Time= 34,16000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,17000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,18000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,19000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,20000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,21000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,22000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,23000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,24000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,25000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,26000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,27000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,28000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,29000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,30000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,31000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,32000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,33000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,34000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,35000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,36000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,37000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,38000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,39000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,40000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,41000, dt= 0,01000 (Iter: 1 => Converg)

Time= 34,42000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,43000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,44000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,45000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,46000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,47000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,48000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,49000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,50000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,51000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,52000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,53000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,54000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,55000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,56000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,57000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,58000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,59000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,60000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,61000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,62000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,63000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,64000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,65000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,66000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,67000, dt= 0,01000 (Iter: 1 => Converg)

Time= 34,68000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,69000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,70000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,71000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,72000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,73000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,74000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,75000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,76000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,77000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,78000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,79000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,80000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,81000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,82000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,83000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,84000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,85000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,86000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,87000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,88000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,89000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,90000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,91000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,92000, dt= 0,01000 (Iter: 1 => Converg)
Time= 34,93000, dt= 0,01000 (Iter: 1 => Converg)

Time= 34,94000, dt= 0,01000 (Iter: 1 => Converg)

Time= 34,95000, dt= 0,01000 (Iter: 1 => Converg)

Time= 34,96000, dt= 0,01000 (Iter: 1 => Converg)

Time= 34,97000, dt= 0,01000 (Iter: 1 => Converg)

Time-history Analysis completed

Total Analysis Time: 0h:13min:7sec

#####

Performance Criteria

Otp No: 977 Time= 9,7600, yield reached. Elm: bmx314.
Reinf. Steel Strain = 0,0025047 - Sec(b)

Otp No: 979 Time= 9,7800, yield reached. Elm: bmx313.
Reinf. Steel Strain = 0,00253492 - Sec(b)

Otp No: 980 Time= 9,7900, yield reached. Elm: bmx314.
Reinf. Steel Strain = 0,00258078 - Sec(a)

Otp No: 981 Time= 9,8000, yield reached. Elm: bmx313.
Reinf. Steel Strain = 0,00251676 - Sec(a)

Otpt No: 981 Time= 9,8000, yield reached. Elm: bmx315.
Reinf. Steel Strain = 0,00254667 - Sec(b)

Otpt No: 985 Time= 9,8400, yield reached. Elm: bmx214.
Reinf. Steel Strain = 0,00252411 - Sec(a)

Otpt No: 985 Time= 9,8400, yield reached. Elm: bmx214.
Reinf. Steel Strain = 0,00251886 - Sec(b)

Otpt No:1015 Time=10,1400, yield reached. Elm: bmx113.
Reinf. Steel Strain = 0,00257522 - Sec(a)

Otpt No:1017 Time=10,1600, yield reached. Elm: bmx113.
Reinf. Steel Strain = 0,00261245 - Sec(b)

Otpt No:1018 Time=10,1700, yield reached. Elm: bmx213.
Reinf. Steel Strain = 0,00251186 - Sec(a)

Otpt No:1018 Time=10,1700, yield reached. Elm: bmx213.
Reinf. Steel Strain = 0,00251044 - Sec(b)

Otpt No:1019 Time=10,1800, yield reached. Elm: bmx114.
Reinf. Steel Strain = 0,00263524 - Sec(a)

Otpt No:1019 Time=10,1800, yield reached. Elm: bmx312.
Reinf. Steel Strain = 0,0026648 - Sec(b)

Otpt No:1020 Time=10,1900, yield reached. Elm: bmx112.
Reinf. Steel Strain = 0,00263634 - Sec(a)

Otpt No:1020 Time=10,1900, yield reached. Elm: bmx114.
Reinf. Steel Strain = 0,00256296 - Sec(b)

Otpt No:1020 Time=10,1900, yield reached. Elm: bmx116.
Reinf. Steel Strain = 0,00257012 - Sec(a)

Otpt No:1021 Time=10,2000, yield reached. Elm: bmx112.
Reinf. Steel Strain = 0,00257738 - Sec(b)

Otpt No:1021 Time=10,2000, yield reached. Elm: bmx212.
Reinf. Steel Strain = 0,00259709 - Sec(b)

Otpt No:1021 Time=10,2000, yield reached. Elm: bmx115.
Reinf. Steel Strain = 0,0028115 - Sec(a)

Otpt No:1021 Time=10,2000, yield reached. Elm: bmx115.
Reinf. Steel Strain = 0,00257686 - Sec(b)

Otpt No:1021 Time=10,2000, yield reached. Elm: bmx116.
Reinf. Steel Strain = 0,0025022 - Sec(b)

Otpt No:1022 Time=10,2100, yield reached. Elm: bmx212.
Reinf. Steel Strain = 0,00253334 - Sec(a)

Otpt No:1022 Time=10,2100, yield reached. Elm: bmx312.
Reinf. Steel Strain = 0,00258612 - Sec(a)

Otpt No:1022 Time=10,2100, yield reached. Elm: bmx117.
Reinf. Steel Strain = 0,00252202 - Sec(a)

Otpt No:1023 Time=10,2200, yield reached. Elm: bmx215.
Reinf. Steel Strain = 0,00262129 - Sec(a)

Otpt No:1023 Time=10,2200, yield reached. Elm: bmx215.
Reinf. Steel Strain = 0,00261249 - Sec(b)

Otpt No:1025 Time=10,2400, yield reached. Elm: bmx216.
Reinf. Steel Strain = 0,00267276 - Sec(a)

Otpt No:1025 Time=10,2400, yield reached. Elm: bmx216.
Reinf. Steel Strain = 0,00266447 - Sec(b)

Otpt No:1025 Time=10,2400, yield reached. Elm: bmx117.
Reinf. Steel Strain = 0,00251393 - Sec(b)

Otpt No:1026 Time=10,2500, yield reached. Elm: bmx315.
Reinf. Steel Strain = 0,0027384 - Sec(a)

Otpt No:1029 Time=10,2800, yield reached. Elm: bmx316.
Reinf. Steel Strain = 0,00251873 - Sec(b)

Otpt No:1033 Time=10,3200, yield reached. Elm: bmx316.
Reinf. Steel Strain = 0,00252611 - Sec(a)

Otpt No:1037 Time=10,3600, yield reached. Elm: col311.
Reinf. Steel Strain = 0,00260851 - Sec(a)

Otpt No:1041 Time=10,4000, yield reached. Elm: col211.
Reinf. Steel Strain = 0,00266261 - Sec(a)

Otpt No:1043 Time=10,4200, crush_unc reached. Elm: col411.
Concrete Strain = -0,00357152 - Sec(a)

Otpt No:1044 Time=10,4300, yield reached. Elm: col212.
Reinf. Steel Strain = 0,00253525 - Sec(a)

Otpt No:1044 Time=10,4300, yield reached. Elm: col312.
Reinf. Steel Strain = 0,00252159 - Sec(a)

Otpt No:1046 Time=10,4500, yield reached. Elm: col411.
Reinf. Steel Strain = 0,00256043 - Sec(a)

Otpt No:1048 Time=10,4700, yield reached. Elm: bmx217.
Reinf. Steel Strain = 0,00253488 - Sec(a)

Otpt No:1049 Time=10,4800, yield reached. Elm: bmx217.
Reinf. Steel Strain = 0,00263935 - Sec(b)

Otpt No:1051 Time=10,5000, yield reached. Elm: bmx118.
Reinf. Steel Strain = 0,00263252 - Sec(a)

Otpt No:1053 Time=10,5200, yield reached. Elm: col111.
Reinf. Steel Strain = 0,00251571 - Sec(a)

Otpt No:1058 Time=10,5700, yield reached. Elm: bmx317.
Reinf. Steel Strain = 0,00254474 - Sec(b)

Otpt No:1060 Time=10,5900, yield reached. Elm: bmx317.
Reinf. Steel Strain = 0,00252848 - Sec(a)

Otpt No:1061 Time=10,6000, yield reached. Elm: bmx118.
Reinf. Steel Strain = 0,00251711 - Sec(b)

Otpt No:1078 Time=10,7700, yield reached. Elm: col215.
Reinf. Steel Strain = 0,00251142 - Sec(b)

Otpt No:1206 Time=12,0500, crush_unc reached. Elm: col111.
Concrete Strain = -0,00352136 - Sec(a)

Otpt No:1224 Time=12,2300, chordRot_Cap reached. Elm: col111.
Chord Rotation Capacity = 0,01552936 - End (A) - axis(2)

Otpt No:1234 Time=12,3300, crush_conf reached. Elm: col111.
Concrete Strain = -0,00802449 - Sec(a)

Otpt No:1234 Time=12,3300, crush_unc reached. Elm: col211.
Concrete Strain = -0,00352025 - Sec(a)

Otpt No:1250 Time=12,4900, yield reached. Elm: col315.
Reinf. Steel Strain = 0,00250556 - Sec(b)

Otpt No:1251 Time=12,5000, yield reached. Elm: bmx318.
Reinf. Steel Strain = 0,00253054 - Sec(b)

Otpt No:1262 Time=12,6100, yield reached. Elm: bmx318.
Reinf. Steel Strain = 0,00251981 - Sec(a)

Otpt No:1268 Time=12,6700, yield reached. Elm: bmx218.
Reinf. Steel Strain = 0,00262844 - Sec(a)

Otpt No:1268 Time=12,6700, yield reached. Elm: bmx218.
Reinf. Steel Strain = 0,00257884 - Sec(b)

Otpt No:1275 Time=12,7400, crush_unc reached. Elm: col311.
Concrete Strain = -0,00352652 - Sec(a)

Otpt No:1320 Time=13,1900, chord_rot reached. Elm: bmx314.
Chord Rotation Capacity = 0,02635199 - End (B) - axis(2)

Otpt No:1321 Time=13,2000, chord_rot reached. Elm: bmx313.
Chord Rotation Capacity = 0,02383013 - End (A) - axis(2)

Otpt No:1322 Time=13,2100, chord_rot reached. Elm: bmx114.
Chord Rotation Capacity = 0,02142952 - End (A) - axis(2)

Otpt No:1323 Time=13,2200, chord_rot reached. Elm: bmx112.
Chord Rotation Capacity = 0,01957158 - End (A) - axis(2)

Otpt No:1323 Time=13,2200, chord_rot reached. Elm: bmx314.
Chord Rotation Capacity = 0,02511718 - End (A) - axis(2)

Otpt No:1324 Time=13,2300, chord_rot reached. Elm: bmx114.
Chord Rotation Capacity = 0,02119342 - End (B) - axis(2)

Otpt No:1326 Time=13,2500, chordRot_Cap reached. Elm: col211.
Chord Rotation Capacity = 0,0153246 - End (A) - axis(2)

Otpt No:1327 Time=13,2600, chordRot_Cap reached. Elm: col311.
Chord Rotation Capacity = 0,01248628 - End (A) - axis(2)

Otpt No:1427 Time=14,2600, chordRot_Cap reached. Elm: col411.
Chord Rotation Capacity = 0,01641904 - End (A) - axis(2)

Otpt No:1433 Time=14,3200, crush_conf reached. Elm: col411.
Concrete Strain = -0,00813434 - Sec(a)

Otpt No:1460 Time=14,5900, yield reached. Elm: col216.
Reinf. Steel Strain = 0,00251508 - Sec(b)

Ek 2: ProBina Analiz Raporu Tamamı

TABLE: Active Degrees of Freedom

UX	UY	UZ	RX	RY	RZ
Yes/No	Yes/No	Yes/No	Yes/No	Yes/No	Yes/No
Yes	Yes	Yes	Yes	Yes	Yes

TABLE: Analysis Options

Solver	SolverProc	Force32Bit	StiffCase	GeomMod
Text	Text	Yes/No	Text	Yes/No
Advanced	Auto	No	None	None

TABLE: Area Loads - Uniform

Area	LoadPat	CoordSys	Dir	UnifLoad
Text	Text	Text	Text	KN/m2
1	D	GLOBAL	Gravity	10
1	H	GLOBAL	Gravity	100
2	D	GLOBAL	Gravity	10
2	H	GLOBAL	Gravity	100
3	D	GLOBAL	Gravity	10
4	H	GLOBAL	Gravity	100
4	D	GLOBAL	Gravity	10
5	H	GLOBAL	Gravity	100
5	D	GLOBAL	Gravity	10
6	H	GLOBAL	Gravity	100
6	D	GLOBAL	Gravity	10
7	D	GLOBAL	Gravity	10
7	H	GLOBAL	Gravity	100
8	D	GLOBAL	Gravity	10
8	H	GLOBAL	Gravity	100
9	D	GLOBAL	Gravity	10
9	H	GLOBAL	Gravity	100
10	H	GLOBAL	Gravity	100
10	D	GLOBAL	Gravity	10
11	H	GLOBAL	Gravity	100
11	D	GLOBAL	Gravity	10
12	H	GLOBAL	Gravity	100
12	D	GLOBAL	Gravity	10
13	D	GLOBAL	Gravity	10
13	H	GLOBAL	Gravity	100
14	D	GLOBAL	Gravity	10
14	H	GLOBAL	Gravity	100
15	D	GLOBAL	Gravity	10
15	H	GLOBAL	Gravity	100

16	H	GLOBAL	Gravity	100
16	D	GLOBAL	Gravity	10
17	H	GLOBAL	Gravity	100
17	D	GLOBAL	Gravity	10
18	H	GLOBAL	Gravity	100
18	D	GLOBAL	Gravity	10
19	S	GLOBAL	Gravity	5
20	S	GLOBAL	Gravity	5
21	S	GLOBAL	Gravity	5
22	S	GLOBAL	Gravity	5
23	S	GLOBAL	Gravity	5
24	S	GLOBAL	Gravity	5
25	S	GLOBAL	Gravity	5
26	S	GLOBAL	Gravity	5
27	S	GLOBAL	Gravity	5
28	S	GLOBAL	Gravity	5
29	S	GLOBAL	Gravity	5
30	S	GLOBAL	Gravity	5
31	S	GLOBAL	Gravity	5
32	S	GLOBAL	Gravity	5
33	S	GLOBAL	Gravity	5
34	S	GLOBAL	Gravity	5
35	S	GLOBAL	Gravity	5
36	S	GLOBAL	Gravity	5
37	S	GLOBAL	Gravity	5
38	S	GLOBAL	Gravity	5
39	S	GLOBAL	Gravity	5
40	S	GLOBAL	Gravity	5
41	S	GLOBAL	Gravity	5
42	S	GLOBAL	Gravity	5
43	S	GLOBAL	Gravity	5
44	S	GLOBAL	Gravity	5
45	S	GLOBAL	Gravity	5
46	S	GLOBAL	Gravity	5
47	S	GLOBAL	Gravity	5
48	S	GLOBAL	Gravity	5
49	S	GLOBAL	Gravity	5
50	S	GLOBAL	Gravity	5
51	S	GLOBAL	Gravity	5
52	S	GLOBAL	Gravity	5
58	S	GLOBAL	Gravity	5
59	S	GLOBAL	Gravity	5
60	S	GLOBAL	Gravity	5
61	S	GLOBAL	Gravity	5
62	S	GLOBAL	Gravity	5
63	S	GLOBAL	Gravity	5

TABLE: Area Section Assignments

Area	Section	MatProp
Text	Text	Text
1	DOSEME	Default
2	DOSEME	Default
3	DOSEME	Default
4	DOSEME	Default
5	DOSEME	Default
6	DOSEME	Default
7	DOSEME	Default
8	DOSEME	Default
9	DOSEME	Default
10	DOSEME	Default
11	DOSEME	Default
12	DOSEME	Default
13	DOSEME	Default
14	DOSEME	Default
15	DOSEME	Default
16	DOSEME	Default
17	DOSEME	Default
18	DOSEME	Default
19	DOSEME	Default
20	DOSEME	Default
21	DOSEME	Default
22	DOSEME	Default
23	DOSEME	Default
24	DOSEME	Default
25	DOSEME	Default
26	DOSEME	Default
27	DOSEME	Default
28	DOSEME	Default
29	DOSEME	Default
30	DOSEME	Default
31	DOSEME	Default
32	DOSEME	Default
33	DOSEME	Default
34	DOSEME	Default
35	DOSEME	Default
36	DOSEME	Default
37	DOSEME	Default
38	DOSEME	Default
39	DOSEME	Default
40	DOSEME	Default
41	DOSEME	Default
42	DOSEME	Default
43	DOSEME	Default

44	DOSEME	Default
45	DOSEME	Default
46	DOSEME	Default
47	DOSEME	Default
48	DOSEME	Default
49	DOSEME	Default
50	DOSEME	Default
51	DOSEME	Default
52	DOSEME	Default
58	Cephe	Default
59	Cephe	Default
60	Cephe	Default
61	Cephe	Default
62	Cephe	Default
63	Cephe	Default

Ek 3: Sap2000 Analiz Raporu Tamamı

TABLE: Area Section Property Design Parameters		
Section	RebarMat	RebarOpt
Text	Text	Text
Cephe	None	Default
DOSEME	S420	Default

TABLE: Area Section Property - Time Dependent			
Section	TypeSize	AutoSFSize	UserValSize
Text	Text	Unitless	m
Cephe	Auto	1	
DOSEME	Auto	1	

TABLE: Area Section Properties								
Section	Material	MatAngle	AreaType	Type	DrillDOF	Thickness	BendThick	Arc
Text	Text	Degrees	Text	Text	Yes/No	m	m	Degrees
Cephe	Cephe	0	Shell	Shell-Thin	Yes	0,05	0,05	
DOSEME	C40/50	0	Shell	Shell-Thick	Yes	0,3	0,3	

InComp	Coord Sys	Color	TotalWt	TotalMass	F11Mod	F22Mod	F12Mod	M11Mod
Yes/No	Text	Text	KN	KN-s2/m	Unitless	Unitless	Unitless	Unitless
		167447	148.438.165.52	151.467.515.84				
		03	3.733	0.543	1	1	1	1
		Magenta	960.016.429.55	978.944.301.32				
		ta	5.919	3.998	1	1	1	1

M22Mod	M12Mod	V13Mod	V23Mod	MMod	WMod	GUID	Notes
Unitless	Unitless	Unitless	Unitless	Unitless	Unitless	Text	Text
1	1	1	1	1	1		Added 18.06.2019 03:14:33
1	1	1	1	1	1		Added 18.06.2019 02:41:31

TABLE: Area Loads - Uniform				
Area	LoadPat	CoordSys	Dir	UnifLoad
Text	Text	Text	Text	KN/m2
1	D	GLOBAL	Gravity	10
1	H	GLOBAL	Gravity	100
2	D	GLOBAL	Gravity	10
2	H	GLOBAL	Gravity	100
3	D	GLOBAL	Gravity	10
4	H	GLOBAL	Gravity	100
4	D	GLOBAL	Gravity	10
5	H	GLOBAL	Gravity	100
5	D	GLOBAL	Gravity	10
6	H	GLOBAL	Gravity	100
6	D	GLOBAL	Gravity	10
7	D	GLOBAL	Gravity	10
7	H	GLOBAL	Gravity	100
8	D	GLOBAL	Gravity	10
8	H	GLOBAL	Gravity	100
9	D	GLOBAL	Gravity	10
9	H	GLOBAL	Gravity	100
10	H	GLOBAL	Gravity	100
10	D	GLOBAL	Gravity	10
11	H	GLOBAL	Gravity	100
11	D	GLOBAL	Gravity	10
12	H	GLOBAL	Gravity	100
12	D	GLOBAL	Gravity	10
13	D	GLOBAL	Gravity	10
13	H	GLOBAL	Gravity	100
14	D	GLOBAL	Gravity	10
14	H	GLOBAL	Gravity	100
15	D	GLOBAL	Gravity	10
15	H	GLOBAL	Gravity	100
16	H	GLOBAL	Gravity	100
16	D	GLOBAL	Gravity	10
17	H	GLOBAL	Gravity	100
17	D	GLOBAL	Gravity	10
18	H	GLOBAL	Gravity	100
18	D	GLOBAL	Gravity	10
19	S	GLOBAL	Gravity	5
20	S	GLOBAL	Gravity	5
21	S	GLOBAL	Gravity	5
22	S	GLOBAL	Gravity	5
23	S	GLOBAL	Gravity	5
24	S	GLOBAL	Gravity	5
25	S	GLOBAL	Gravity	5
26	S	GLOBAL	Gravity	5

27	S	GLOBAL	Gravity	5
28	S	GLOBAL	Gravity	5
29	S	GLOBAL	Gravity	5
30	S	GLOBAL	Gravity	5
31	S	GLOBAL	Gravity	5
32	S	GLOBAL	Gravity	5
33	S	GLOBAL	Gravity	5
34	S	GLOBAL	Gravity	5
35	S	GLOBAL	Gravity	5
36	S	GLOBAL	Gravity	5
37	S	GLOBAL	Gravity	5
38	S	GLOBAL	Gravity	5
39	S	GLOBAL	Gravity	5
40	S	GLOBAL	Gravity	5
41	S	GLOBAL	Gravity	5
42	S	GLOBAL	Gravity	5
43	S	GLOBAL	Gravity	5
44	S	GLOBAL	Gravity	5
45	S	GLOBAL	Gravity	5
46	S	GLOBAL	Gravity	5
47	S	GLOBAL	Gravity	5
48	S	GLOBAL	Gravity	5
49	S	GLOBAL	Gravity	5
50	S	GLOBAL	Gravity	5
51	S	GLOBAL	Gravity	5
52	S	GLOBAL	Gravity	5
58	S	GLOBAL	Gravity	5
59	S	GLOBAL	Gravity	5
60	S	GLOBAL	Gravity	5
61	S	GLOBAL	Gravity	5
62	S	GLOBAL	Gravity	5
63	S	GLOBAL	Gravity	5

TABLE: Area Section Assignments		
Area	Section	MatProp
Text	Text	Text
1	DOSEME	Default
2	DOSEME	Default
3	DOSEME	Default
4	DOSEME	Default
5	DOSEME	Default
6	DOSEME	Default
7	DOSEME	Default
8	DOSEME	Default
9	DOSEME	Default

10	DOSEME	Default
11	DOSEME	Default
12	DOSEME	Default
13	DOSEME	Default
14	DOSEME	Default
15	DOSEME	Default
16	DOSEME	Default
17	DOSEME	Default
18	DOSEME	Default
19	DOSEME	Default
20	DOSEME	Default
21	DOSEME	Default
22	DOSEME	Default
23	DOSEME	Default
24	DOSEME	Default
25	DOSEME	Default
26	DOSEME	Default
27	DOSEME	Default
28	DOSEME	Default
29	DOSEME	Default
30	DOSEME	Default
31	DOSEME	Default
32	DOSEME	Default
33	DOSEME	Default
34	DOSEME	Default
35	DOSEME	Default
36	DOSEME	Default
37	DOSEME	Default
38	DOSEME	Default
39	DOSEME	Default
40	DOSEME	Default
41	DOSEME	Default
42	DOSEME	Default
43	DOSEME	Default
44	DOSEME	Default
45	DOSEME	Default
46	DOSEME	Default
47	DOSEME	Default
48	DOSEME	Default
49	DOSEME	Default
50	DOSEME	Default
51	DOSEME	Default
52	DOSEME	Default
58	Cephe	Default
59	Cephe	Default
60	Cephe	Default

61	Cephe	Default
62	Cephe	Default
63	Cephe	Default

TABLE: Area Section Property Design Parameters

Section	RebarMat	RebarOpt
Text	Text	Text
Cephe	None	Default
DOSEME	S420	Default

TABLE: Area Section Property - Time Dependent

Section	TypeSize	AutoSFSize	UserValSize
Text	Text	Unitless	m
Cephe	Auto	1	
DOSEME	Auto	1	

TABLE: Auto Seismic - TSC 2007

LoadP at	Dir	Percent Ecc	EccOve rride	Period Calc	Ct	Use rZ	ZCo de	Seismic Zone	SiteC lass	A 0	I	R
Text	Text	Unitless	Yes/No	Text	Unitl ess	Yes/ No	Text	Text	Text	Text	Unitl ess	Unitl ess
E	X	0,05	No	Prog Calc		No	0,3 6	Zone 1	Z2	0, 4	1	8

TABLE: Auto Wave 3 - Wave Characteristics - General

WaveCha r	WaveType	KinFacto r	SWaterDept h	WaveHeigh t	WavePerio d	WaveTheor y
Text	Text	Unitless	m	m	Sec	Text
Default	From Theory	1	45	18	12	Linear

TABLE: Auto Wind - TS 498-97

LoadP at	ExposeFr om	Angle	Windward Cp	Leeward Cp	UserZ	WindSpe ed	CoeffD yn	ExpWidth
Text	Text	Degre es	Unitless	Unitless	Yes/N o	meter/se c	Text	Text
S	Diaphrag ms	0	0,8	0,5	No	30	1	From diaphs

TABLE: Auto Wind Exposure For Horizontal Diaphragms

LoadPat	Diaphragm	DiaphragmZ	X	Y	TribWidth	TribDepth	TribHeight
Text	Text	m	m	m	m	m	m

S

TABLE: Case - Modal 1 - General

Case	ModeType	MaxNumModes	MinNumModes	EigenShift	EigenCutoff	EigenTol	AutoShift
Text	Text	Unitless	Unitless	Cyc/sec	Cyc/sec	Unitless	Text

MODAL

L Eigen 12 1 0 0 1,00E-09 Yes

TABLE: Case - Static 1 - Load Assignments

Case	LoadType	LoadName	LoadSF
Text	Text	Text	Unitless

DEAD	Load pattern	D	1
H	Load pattern	H	1
Kar	Load pattern	S	1
E	Load pattern	E	1

**TABLE:
Combination
Definitions**

Comb oNa me	Com boTy pe	Auto Desi gn	CaseT ype	Cas eN ame	Scal eFac tor	Steel Desi gn	Conc Desi gn	Alum Desi gn	Cold Desi gn	GUID	N otes
Text	Text	Yes/ No	Text	Text	Unitl ess	Text	Text	Text	Text	Text	Text
14D+	Linea		Linear	DE		Non	Non		Non	019f6993-b654-4a2d-838e-684d8624bcee	
16L	r Add	No	Static	AD	14	e	e	None	e		
14D+			Linear								
16L			Static	H	16						
D+L+	Linea		Linear	DE		Non	Non		Non	019f6993-b654-4a2d-838e-684d8624bcee	
E	r Add	No	Static	AD	1	e	e	None	e		
D+L+			Linear								
E			Static	E	1						
D+L+			Linear								
E			Static	H	1						
D+L+	Linea		Linear	DE		Non	Non		Non	019f6993-b654-4a2d-838e-684d8624bcee	
S	r Add	No	Static	AD	1	e	e	None	e		
D+L+			Linear								
S			Static	H	1						
D+L+			Linear								
S			Static	Kar	1						

**TABLE:
Connectivity -
Area**

Ar ea	NumJ oints	Joi nt1	Joi nt 2	Joi nt 3	Joi nt 4	Peri met er	Area Area	Volu me	Ce ntr oid X	Cent roid Y	Cent roid Z	GUID
Text	Unitle ss	Text	Text	Text	Text	m	m2	m3	m	m	m	Text
1	4	49	64	88	76	24	36	10,8	3	3	0	b1930d1b-688e-400c-aeab-c9becb273855
2	4	64	70	94	88	24	36	10,8	9	3	0	43ec407f-4246-4d8b-a45e-cf29f768a987
3	4	70	55	82	94	24	36	10,8	15	3	0	3fe21929-ecf6-453c-8e7f-845141c3677a
4	4	50	65	89	77	24	36	10,8	3	3	3,1	7b8004f7-c214-40e0-

												bd82-de8421d176bc
5	4	65	71	95	89	24	36	10,8	9	3	3,1	b73e4ffd-506a-4060- bccb-2664c901efb8
6	4	71	56	83	95	24	36	10,8	15	3	3,1	087caf6e-ae79-4c5a- 98cb-6dbe7bea65f0
7	4	51	66	90	78	24	36	10,8	3	3	6,2	38aef03d-466d- 4295-b931- 17bdbe340fa2
8	4	66	72	96	90	24	36	10,8	9	3	6,2	e49d956f-c6d4-49dc- 9f48-4317c4abfc5e
9	4	72	57	84	96	24	36	10,8	15	3	6,2	a89e8c97-e256- 41ad-8180- e7972f2b8098
10	4	52	67	91	79	24	36	10,8	3	3	9,3	2414050d-66eb- 4654-8384- 15f3bb7f4bf9
11	4	67	73	97	91	24	36	10,8	9	3	9,3	f7ca5a6e-ee66-42f3- bf62-b14887d937a3
12	4	73	58	85	97	24	36	10,8	15	3	9,3	fcaed6c1-8f10-4590- 98c5-cc71d2b683ee
13	4	53	68	92	80	24	36	10,8	3	3	12,4	ad3a4346-6a00- 4453-9737- 7f9b840822dc
14	4	68	74	98	92	24	36	10,8	9	3	12,4	c8edae93-f0ae-4f0e- 96ec-66d60369c5ad
15	4	74	59	86	98	24	36	10,8	15	3	12,4	71cf5c68-a4bd-48a5- b47b-114cd4d3960a
16	4	54	69	93	81	24	36	10,8	3	3	15,5	98a0f79b-658e- 44bb-bfc7- 9377ab6bef30
17	4	69	75	99	93	24	36	10,8	9	3	15,5	5ac94552-df4b-4b38- 8936-a0da2e4f393f
18	4	75	60	87	99	24	36	10,8	15	3	15,5	96116cae-ea77-4c45- a87c-02035dae21f0
19	4	49	64	65	50	18,2	18,6	5,58	3	0	1,55	6894b228-ff44-4418- b9ff-4b075576c973
20	4	64	70	71	65	18,2	18,6	5,58	9	0	1,55	531d0c28-dc2d- 4d49-a725- bc95538b19e0
21	4	70	55	56	71	18,2	18,6	5,58	15	0	1,55	84fb2b83-1c90-4357- a366-66e4ebf11dcd
22	4	50	65	66	51	18,2	18,6	5,58	3	0	4,65	3a4135a6-9d3b- 44d8-bf2e- b1f2bdc2733f
23	4	65	71	72	66	18,2	18,6	5,58	9	0	4,65	8bec98c8-ed73- 4e7d-b400- d0c399be3ae7
24	4	71	56	57	72	18,2	18,6	5,58	15	0	4,65	48e2f771-9c5b-4a8c- 872f-2db983e167d3
25	4	51	66	67	52	18,2	18,6	5,58	3	0	7,75	4b1ad9c7-4a29- 4d2b-8e1c- 65c7b0ee57ca

26	4	66	72	73	67	18,2	18,6	5,58	9	0	7,75	9b64295d-d187-48b1-8fdd-beaee782d9743d1e2014-c403-42e1-9584-
27	4	72	57	58	73	18,2	18,6	5,58	15	0	7,75	8305769bf5bd10,8
28	4	52	67	68	53	18,2	18,6	5,58	3	0	5	8dad2e0d-ff81-44c4-964a-4b777c44c0d388935587-d9d4-4afb-abe1-ab3a157b1607d7f3957b-eb1a-40b3-802d-2cab37365e68e8810a6d-a37b-4b2a-b996-8f7dc48cc71e
29	4	67	73	74	68	18,2	18,6	5,58	9	0	5	585ef81f-186d-45eb-8319-ecf36757e4e68bdb246b-b7b7-4077-8f3a-cdeccd13febfc2ab24c03-3428-4f9c-8df7-3f4bd252fdfd623d9566-61ee-4adf-b8ce-588c5ccd5b515d149ddc-454e-4bed-b368-45dfbbb87f21696f3202-cfce-428c-b857-30414e5aa3ffd92b1f83-8904-4a27-b2dd-9b68d1b8e4bcce3a2e2b-ee6c-49fd-8b26-666e409314446c19a3e0-e700-4413-8ee4-6f532d6627d5d07daf70-21e1-4df8-ada0-33cfa79730d3afb236ce-193f-4bc1-b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
30	4	73	58	59	74	18,2	18,6	5,58	15	0	5	964a-4b777c44c0d388935587-d9d4-4afb-abe1-ab3a157b1607d7f3957b-eb1a-40b3-802d-2cab37365e68e8810a6d-a37b-4b2a-b996-8f7dc48cc71e
31	4	53	68	69	54	18,2	18,6	5,58	3	0	5	585ef81f-186d-45eb-8319-ecf36757e4e68bdb246b-b7b7-4077-8f3a-cdeccd13febfc2ab24c03-3428-4f9c-8df7-3f4bd252fdfd623d9566-61ee-4adf-b8ce-588c5ccd5b515d149ddc-454e-4bed-b368-45dfbbb87f21696f3202-cfce-428c-b857-30414e5aa3ffd92b1f83-8904-4a27-b2dd-9b68d1b8e4bcce3a2e2b-ee6c-49fd-8b26-666e409314446c19a3e0-e700-4413-8ee4-6f532d6627d5d07daf70-21e1-4df8-ada0-33cfa79730d3afb236ce-193f-4bc1-b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
32	4	68	74	75	69	18,2	18,6	5,58	9	0	5	8319-ecf36757e4e68bdb246b-b7b7-4077-8f3a-cdeccd13febfc2ab24c03-3428-4f9c-8df7-3f4bd252fdfd623d9566-61ee-4adf-b8ce-588c5ccd5b515d149ddc-454e-4bed-b368-45dfbbb87f21696f3202-cfce-428c-b857-30414e5aa3ffd92b1f83-8904-4a27-b2dd-9b68d1b8e4bcce3a2e2b-ee6c-49fd-8b26-666e409314446c19a3e0-e700-4413-8ee4-6f532d6627d5d07daf70-21e1-4df8-ada0-33cfa79730d3afb236ce-193f-4bc1-b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
33	4	74	59	60	75	18,2	18,6	5,58	15	0	5	8319-ecf36757e4e68bdb246b-b7b7-4077-8f3a-cdeccd13febfc2ab24c03-3428-4f9c-8df7-3f4bd252fdfd623d9566-61ee-4adf-b8ce-588c5ccd5b515d149ddc-454e-4bed-b368-45dfbbb87f21696f3202-cfce-428c-b857-30414e5aa3ffd92b1f83-8904-4a27-b2dd-9b68d1b8e4bcce3a2e2b-ee6c-49fd-8b26-666e409314446c19a3e0-e700-4413-8ee4-6f532d6627d5d07daf70-21e1-4df8-ada0-33cfa79730d3afb236ce-193f-4bc1-b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
34	4	76	88	89	77	18,2	18,6	5,58	3	6	1,55	8df7-3f4bd252fdfd623d9566-61ee-4adf-b8ce-588c5ccd5b515d149ddc-454e-4bed-b368-45dfbbb87f21696f3202-cfce-428c-b857-30414e5aa3ffd92b1f83-8904-4a27-b2dd-9b68d1b8e4bcce3a2e2b-ee6c-49fd-8b26-666e409314446c19a3e0-e700-4413-8ee4-6f532d6627d5d07daf70-21e1-4df8-ada0-33cfa79730d3afb236ce-193f-4bc1-b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
35	4	88	94	95	89	18,2	18,6	5,58	9	6	1,55	b8ce-588c5ccd5b515d149ddc-454e-4bed-b368-45dfbbb87f21696f3202-cfce-428c-b857-30414e5aa3ffd92b1f83-8904-4a27-b2dd-9b68d1b8e4bcce3a2e2b-ee6c-49fd-8b26-666e409314446c19a3e0-e700-4413-8ee4-6f532d6627d5d07daf70-21e1-4df8-ada0-33cfa79730d3afb236ce-193f-4bc1-b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
36	4	94	82	83	95	18,2	18,6	5,58	15	6	1,55	45dfbbb87f21696f3202-cfce-428c-b857-30414e5aa3ffd92b1f83-8904-4a27-b2dd-9b68d1b8e4bcce3a2e2b-ee6c-49fd-8b26-666e409314446c19a3e0-e700-4413-8ee4-6f532d6627d5d07daf70-21e1-4df8-ada0-33cfa79730d3afb236ce-193f-4bc1-b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
37	4	77	89	90	78	18,2	18,6	5,58	3	6	4,65	b857-30414e5aa3ffd92b1f83-8904-4a27-b2dd-9b68d1b8e4bcce3a2e2b-ee6c-49fd-8b26-666e409314446c19a3e0-e700-4413-8ee4-6f532d6627d5d07daf70-21e1-4df8-ada0-33cfa79730d3afb236ce-193f-4bc1-b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
38	4	89	95	96	90	18,2	18,6	5,58	9	6	4,65	9b68d1b8e4bcce3a2e2b-ee6c-49fd-8b26-666e409314446c19a3e0-e700-4413-8ee4-6f532d6627d5d07daf70-21e1-4df8-ada0-33cfa79730d3afb236ce-193f-4bc1-b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
39	4	95	83	84	96	18,2	18,6	5,58	15	6	4,65	8b26-666e409314446c19a3e0-e700-4413-8ee4-6f532d6627d5d07daf70-21e1-4df8-ada0-33cfa79730d3afb236ce-193f-4bc1-b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
40	4	78	90	91	79	18,2	18,6	5,58	3	6	7,75	6f532d6627d5d07daf70-21e1-4df8-ada0-33cfa79730d3afb236ce-193f-4bc1-b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
41	4	90	96	97	91	18,2	18,6	5,58	9	6	7,75	ada0-33cfa79730d3afb236ce-193f-4bc1-b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
42	4	96	84	85	97	18,2	18,6	5,58	15	6	7,75	b48c-6ccf1a6defa330389d7d-f81c-47d8-a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
43	4	79	91	92	80	18,2	18,6	5,58	3	6	5	a444-dbcf702129292ae623d9-f85c-4ac4-a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
44	4	91	97	98	92	18,2	18,6	5,58	9	6	5	a467-f53967bc315826632013-481c-4b85-9fab-41bfd9736d44a1d2e130-316e-4c1c-be47-
45	4	97	85	86	98	18,2	18,6	5,58	15	6	5	41bfd9736d44a1d2e130-316e-4c1c-be47-
46	4	80	92	93	81	18,2	18,6	5,58	3	6	5	4c1c-be47-

47	4	92	98	99	93	18,2	18,6	5,58	9	6	5	30cefd6b94e4 70232d20-2697- 13,9 4096-bdfe- 032a2bcaa829 83314d00-56e9-
48	4	98	86	87	99	18,2	18,6	5,58	15	6	5	13,9 4680-af9e- 734dbd3fdbdc fe7394e6-bcb4-4b2c-
49	4	49	76	77	50	18,2	18,6	5,58	0	3	1,55	afb4-58da644752d2 8d1b8c76-727c- 4075-9f80-
50	4	50	77	78	51	18,2	18,6	5,58	0	3	4,65	3dc0a1f0f103 904b3a42-fcc0-4c65-
51	4	51	78	79	52	18,2	18,6	5,58	0	3	7,75	a0a2-773ff915b79c 10,8 6eae176d-790f-45d4-
52	4	52	79	80	53	18,2	18,6	5,58	0	3	5	b560-2f2c19abf7d9 42d81991-7976-
58	4	53	80	81	54	18,2	18,6	0,93	0	3	5	13,9 4371-b48b- a3d6f55e01a4 e69f7573-20b9-417c-
59	4	55	82	83	56	18,2	18,6	0,93	18	3	1,55	8723-fa0bcdbfaf6c 42e9331b-7a5f-468a-
60	4	56	83	84	57	18,2	18,6	0,93	18	3	4,65	8655-a7c09f2ca337 fb684320-1a45-4a4e-
61	4	57	84	85	58	18,2	18,6	0,93	18	3	7,75	b9ac-918d032fc9ef 107e86b6-f9be-
62	4	58	85	86	59	18,2	18,6	0,93	18	3	5	10,8 4089-8366- e13f909120ca
63	4	59	86	87	60	18,2	18,6	0,93	18	3	5	13,9 266a7fae-2492-45ab- a5d6-323372801245

**TABLE:
Connectivity -
Frame**

Frame	Joint I	Joint J	IsCurved	Length	Centroid X	Centroid Y	Centroid Z	GUID
Text	Text	Text	Yes/ No	m	m	m	m	Text
1	49	65	No	675.351.760. 196.122	3	0	1,55	1582471d-a25f-46bc- 8faa-63a57c514eb0
2	65	51	No	675.351.760. 196.122	3	0	4,65	24e93f8d-0c57-41dd- 88c5-df563eb6249d
3	51	67	No	675.351.760. 196.122	3	0	7,75	b13c8da6-e281-49de- ba28-3f28965abb74
4	67	53	No	675.351.760. 196.122	3	0	10,85	2a405c8b-5254-47a5- afc5-11192476a3b6
5	53	69	No	675.351.760. 196.122	3	0	13,95	4a3f4b01-4001-4ac4- ac9f-f5e39647f357
6	70	65	No	675.351.760.	9	0	1,55	fd8b9bd3-1d90-4190-

				196.122				8293-4dc4f1e4f464
				675.351.760.				bc3ad7ab-3e9a-4c47-
7	65	72	No	196.122	9	0	4,65	9191-a33d687423c8
				675.351.760.				1c16fe2f-fa77-4652-ac02-
8	72	67	No	196.122	9	0	7,75	9772ed84e0c3
				675.351.760.				2963a17a-c25c-4cfd-
9	67	74	No	196.122	9	0	10,85	9258-2dce28d98571
				675.351.760.				055f9b85-eab3-4db5-
10	74	69	No	196.122	9	0	13,95	a64e-b81a7ae9dd7d
				675.351.760.				2ca6292a-40bd-48c0-
11	70	56	No	196.122	15	0	1,55	8915-7f5950cec32a
				675.351.760.				e07450cb-0d98-4a1c-
12	56	72	No	196.122	15	0	4,65	854f-fb57c0f9928d
				675.351.760.				dd2c9ccb-430f-40e9-
13	72	58	No	196.122	15	0	7,75	9fae-2851c5e6ca13
				675.351.760.				cd293ba5-dc21-4199-
14	58	74	No	196.122	15	0	10,85	bbef-a388fa253f3e
				675.351.760.				53515f8d-f8eb-4bf0-
15	74	60	No	196.122	15	0	13,95	9203-1eb9901cca60
				675.351.760.				0ca8cd5e-5492-4bf5-
16	88	77	No	196.122	3	6	1,55	bb31-6a769fde476d
				675.351.760.				570e1645-61bb-4e1c-
17	77	90	No	196.122	3	6	4,65	9b4a-5532a27828b5
				675.351.760.				bb4e6a85-ea5b-4dc1-
18	90	79	No	196.122	3	6	7,75	846b-f85e4f70a728
				675.351.760.				ec86ed40-373a-4356-
19	79	92	No	196.122	3	6	10,85	b8e8-4cc41b27910a
				675.351.760.				26087082-890b-4027-
20	92	81	No	196.122	3	6	13,95	9618-d04b4160ec10
				675.351.760.				2e8f7234-8799-486c-
21	88	95	No	196.122	9	6	1,55	bcf6-cda4ed11cb45
				675.351.760.				90b703a0-c669-44d6-
22	95	90	No	196.122	9	6	4,65	87aa-33fad882ee1c
				675.351.760.				4af67b18-801e-4394-
23	90	97	No	196.122	9	6	7,75	b4e7-498ae4221db8
				675.351.760.				1bc47154-f25a-42d6-
24	97	92	No	196.122	9	6	10,85	9cc6-d8c01bd5ff5c
				675.351.760.				7b1506e2-3eb7-4a8b-
25	92	99	No	196.122	9	6	13,95	ab74-563f4f8ba6ff
				675.351.760.				5d1d82bf-8146-4b93-
26	82	95	No	196.122	15	6	1,55	9b60-a27bc0df67a7
				675.351.760.				18157a91-03d8-4f95-
27	95	84	No	196.122	15	6	4,65	8ec9-3b746fc5bc99
				675.351.760.				3d158848-568c-4769-
28	84	97	No	196.122	15	6	7,75	9f35-18b003f7bd42
				675.351.760.				c0a0b7df-6719-457b-
29	97	86	No	196.122	15	6	10,85	a2d0-595750cabb9e
				675.351.760.				e8a9a0a4-9744-47ae-
30	86	99	No	196.122	15	6	13,95	8fba-87067605f426
				86.278.618.4				ce7c3e2f-1879-4048-
31	76	51	No	40.492	0	3	3,1	bfe2-d8f580b1d388

32	51	79	No	675.351.760. 196.122	0	3	7,75	a15725f4-73c4-41f8- b461-ed7640571d44
33	79	54	No	86.278.618.4 40.492	0	3	12,4	dc94b900-9dee-4d08- a54b-e0db108f7146
34	55	84	No	86.278.618.4 40.492	18	3	3,1	fb9c93472-30e1-4ce0- a411-0b0c27cb527d
35	84	58	No	675.351.760. 196.122	18	3	7,75	e9870c46-faf4-4ac6- bb50-ff4b264a7c37
36	58	87	No	86.278.618.4 40.492	18	3	12,4	d2c7ada8-b3e6-4de4- 92ad-10db6a0a1d69
114	49	50	No	3,1	0	0	1,55	e1421820-a710-4df5- a2fc-e0659b874c91
115	50	51	No	3,1	0	0	4,65	1d815871-8c73-4b53- 8b7b-3fa25963252b
116	51	52	No	3,1	0	0	7,75	4b97e8a0-e868-4c07- 9f96-1099cb06456f
117	52	53	No	3,1	0	0	10,85	7b0e8c4f-e382-4e34- 9d15-ce0c477dd21f
118	53	54	No	3,1	0	0	13,95	0d663065-f0fa-42e1- ad91-37ecf45347b3
119	55	56	No	3,1	18	0	1,55	3b768ff7-82aa-44ec- 95da-c08a13ada906
120	56	57	No	3,1	18	0	4,65	8461c554-18b1-4757- a99b-8eb05405579c
121	57	58	No	3,1	18	0	7,75	fad96421-10b1-4b54- 898a-43f6bc4367d2
122	58	59	No	3,1	18	0	10,85	0bd77609-0de6-4c08- ac90-60a6266d84dd
123	59	60	No	3,1	18	0	13,95	714670c3-ee60-4975- 8af7-3cade3ead04a
126	64	65	No	3,1	6	0	1,55	e272205d-77f5-45f0- 9701-f31253735832
127	65	66	No	3,1	6	0	4,65	5c0694a4-5445-4d63- a024-dbb48caea7be
128	66	67	No	3,1	6	0	7,75	9b47846e-0ae5-42a3- 8b63-571f5c025330
129	67	68	No	3,1	6	0	10,85	f3e04d34-6048-44bb- 994d-8cc24cf0367c
130	68	69	No	3,1	6	0	13,95	78c72c39-2aee-4ad3- b51a-f52267781351
131	70	71	No	3,1	12	0	1,55	aea7a985-536b-47ab- b40e-163de111eb23
132	71	72	No	3,1	12	0	4,65	98469530-f1d7-4b6f- a4d8-a29dbfdda755
133	72	73	No	3,1	12	0	7,75	540b74aa-6fa9-4304- b844-2ea2451c8bcd
134	73	74	No	3,1	12	0	10,85	e73bdc69-226e-469f- a961-ab82d17d2dd1
135	74	75	No	3,1	12	0	13,95	62733d51-b0fe-424f- 942f-87e5f00a2a95
137	50	65	No	6	3	0	3,1	81297d6f-48a8-49b1- ba29-4ad8fe46440c

138	51	66	No	6	3	0	6,2	4a1466a8-4790-4a4c-add8-8fa842ec1c11bf290a04-1663-4a13-9de9-373ff4fe6c9123ce6a09-300a-46ce-af2-467a7276819ce0782930-1156-4f91-9eaa-534bfb8a286fce5cd3b9-3137-4ef5-b95b-b01dd9f6d6e0aea433f4-ee18-489a-b5b2-3fda25fc27421b1964da-f3d1-45ffa14b-d6ed96478d5c0d044e9e-b28f-4589-87bf-23bc0facba284cf6f8ba-37b2-44ab-91cd-b42c19309cfc83d457f9-87f9-4463-ac15-cd1ebbd122709c329f6f-13a7-41f3-a012-6bbf37ad79a5e617c5c4-a9ba-48d4-befe-e77f899df3c90cf81140-5cea-4aa8-afd4-614ecd427f72f032f49d-60b0-400f-85cf-06938cbb5e8caa4f2977-ade1-40c7-9f6a-1f635fd835487d08b9b9-f267-42b2-a549-69a30bd83520aea93a88-be48-4f6f-b89d-34e1eeb357238c255223-32b5-42c8-9db5-8101ee4412f254f71e9c-7616-429d-87b7-97266e4d7cba676af72d-bdbd-4f12-ab1b-37c8275fcf9e84a3baeb-b272-416b-bf2b-7af551e6589737281e5a-eeefc-4029-b46d-8dd10643c1a4c3a3d21e-2902-4dfb-bc55-8a4ce5377ecfa51c551d-9cdb-4ffe-b0db-453f1d08da2dcf55be1e-58f7-4303-935d-5f2a3b281851f2ba855e-e21d-435e-a69a-0a1c1bcb01f7
139	52	67	No	6	3	0	9,3	
140	53	68	No	6	3	0	12,4	
141	54	69	No	6	3	0	15,5	
143	71	56	No	6	15	0	3,1	
144	72	57	No	6	15	0	6,2	
145	73	58	No	6	15	0	9,3	
146	74	59	No	6	15	0	12,4	
147	75	60	No	6	15	0	15,5	
149	65	71	No	6	9	0	3,1	
150	66	72	No	6	9	0	6,2	
151	67	73	No	6	9	0	9,3	
152	68	74	No	6	9	0	12,4	
153	69	75	No	6	9	0	15,5	
154	76	77	No	3,1	0	6	1,55	
155	77	78	No	3,1	0	6	4,65	
156	78	79	No	3,1	0	6	7,75	
157	79	80	No	3,1	0	6	10,85	
158	80	81	No	3,1	0	6	13,95	
159	82	83	No	3,1	18	6	1,55	
160	83	84	No	3,1	18	6	4,65	
161	84	85	No	3,1	18	6	7,75	
162	85	86	No	3,1	18	6	10,85	
163	86	87	No	3,1	18	6	13,95	
164	88	89	No	3,1	6	6	1,55	
165	89	90	No	3,1	6	6	4,65	

166	90	91	No	3,1	6	6	7,75	eae00168-ea0f-4afc-8467-eeb99c22b940ebe0e27f-2491-4764-
167	91	92	No	3,1	6	6	10,85	8a52-95a7d2afd015fa387da4-e982-479e-
168	92	93	No	3,1	6	6	13,95	885d-413e59d29f9f4d8f0de0-af93-4518-
169	94	95	No	3,1	12	6	1,55	a084-a3a91505d27d90cd226c-f92f-4a93-
170	95	96	No	3,1	12	6	4,65	86a2-3a18506a8b0241f7af49-f745-4875-
171	96	97	No	3,1	12	6	7,75	b12b-8ae424d6c9a91d5e72ec-0647-4623-
172	97	98	No	3,1	12	6	10,85	8485-d74ee357405df7c88f01-6190-4d68-
173	98	99	No	3,1	12	6	13,95	806c-0196963d67f9c91a7908-76dc-43dd-
175	77	89	No	6	3	6	3,1	9ffe-f6a543fa5c4d57921b84-6cb4-40b4-
176	78	90	No	6	3	6	6,2	824b-7973c022b822929b07ef-f904-4dca-
177	79	91	No	6	3	6	9,3	b007-c6230490dcb30f105e66-3712-43f2-
178	80	92	No	6	3	6	12,4	a0f0-45e8a55123fb4397b65a-303d-4d75-
179	81	93	No	6	3	6	15,5	a4bf-c3f4c7488289597bdc75-ddfc-42f8-
181	95	83	No	6	15	6	3,1	9209-eb508edfadbd4d281650d-d828-46ea-
182	96	84	No	6	15	6	6,2	a452-98a17fe71f36a33f2150-bb38-4ff2-
183	97	85	No	6	15	6	9,3	995d-04e5f4925b7881156a0d-9d25-4d0b-
184	98	86	No	6	15	6	12,4	abf9-ae4e9ea5260cbfa10e0b-1a1c-4317-
185	99	87	No	6	15	6	15,5	90dd-3965236dfa364be37cb3-0b31-40c2-
187	89	95	No	6	9	6	3,1	ac18-9ae551c76edfbcf553d0-d293-47ca-
188	90	96	No	6	9	6	6,2	a1b6-3a6cfa0c24059068864b-8b30-48e0-
189	91	97	No	6	9	6	9,3	89e8-35c6f062b44cefafa6e3-6386-43ea-
190	92	98	No	6	9	6	12,4	8512-eeff9d3eca3ae6bc370f-5ee9-48d4-
191	93	99	No	6	9	6	15,5	924f-3205c923bcb6ac3e0e78-03a1-43a9-
196	50	77	No	6	0	3	3,1	b152-ddeeac62aaf1189b86f5-0d2b-47fc-
197	56	83	No	6	18	3	3,1	bd8f-30ec34acfb751cbc65c2-c27d-4bb8-
198	71	95	No	6	12	3	3,1	9402-de118e5d66b0

199	65	89	No	6	6	3	3,1	97bf4feb-d8aa-4c5c-aa57-8d8cfefd1523
200	52	79	No	6	0	3	9,3	2dca1705-3dea-4a74-9445-4ecdc112d847
201	58	85	No	6	18	3	9,3	053a0f85-3ef2-45f4-880d-1fee29cbf44d
202	73	97	No	6	12	3	9,3	dac616f8-e21f-4c1d-88e3-271b28ccaaac
203	67	91	No	6	6	3	9,3	8aad1616-743b-434f-a29e-42329df359f1
204	53	80	No	6	0	3	12,4	cc089108-2427-4104-a14d-d6800bc91204
205	59	86	No	6	18	3	12,4	9a0bf200-f2a6-438b-8726-20d5a7704a0b
206	74	98	No	6	12	3	12,4	d8b6bc13-a682-4416-bfa8-6631cc6d26d3
207	68	92	No	6	6	3	12,4	727ec48c-0857-430c-af11-ce6e538d78cb
208	51	78	No	6	0	3	6,2	f53f1b1e-a28f-4fba-889d-62f174ea1895
209	57	84	No	6	18	3	6,2	6c1d3943-200c-479f-ba7d-770c51f5ed13
210	72	96	No	6	12	3	6,2	301d1825-f507-4b82-b880-c56b07b3c8ba
211	66	90	No	6	6	3	6,2	919e4872-b9e9-4bf0-8288-5c2cdeb940f7
212	54	81	No	6	0	3	15,5	18cf4f3e-5348-460d-a3f1-d0486513122d
213	60	87	No	6	18	3	15,5	396f6fb7-b09a-4528-94e7-ad9f2b56b605
214	75	99	No	6	12	3	15,5	4549fb6d-abd3-4cd9-9ea4-f35267c74c3a
215	69	93	No	6	6	3	15,5	b55b2360-a512-4ae6-972b-f2d3bbf8006d

TABLE: Coordinate Systems

Name	Type	X	Y	Z	AboutZ	AboutY	AboutX
Text	Text	m	m	m	Degrees	Degrees	Degrees
GLOBAL	Cartesian	0 0	0	0	0	0	0

TABLE: Database Format Types

UnitsCurr	OverrideE
Yes/No	Yes/No
Yes	No

TABLE: Frame Auto Mesh Assignments

Frame	AutoMesh	AtJoints	AtFrames	NumSegments	MaxLength	MaxDegrees
Text	Yes/No	Yes/No	Yes/No	Unitless	m	Degrees
1	Yes	Yes	No	0	0	0
2	Yes	Yes	No	0	0	0
3	Yes	Yes	No	0	0	0
4	Yes	Yes	No	0	0	0
5	Yes	Yes	No	0	0	0
6	Yes	Yes	No	0	0	0
7	Yes	Yes	No	0	0	0
8	Yes	Yes	No	0	0	0
9	Yes	Yes	No	0	0	0
10	Yes	Yes	No	0	0	0
11	Yes	Yes	No	0	0	0
12	Yes	Yes	No	0	0	0
13	Yes	Yes	No	0	0	0
14	Yes	Yes	No	0	0	0
15	Yes	Yes	No	0	0	0
16	Yes	Yes	No	0	0	0
17	Yes	Yes	No	0	0	0
18	Yes	Yes	No	0	0	0
19	Yes	Yes	No	0	0	0
20	Yes	Yes	No	0	0	0
21	Yes	Yes	No	0	0	0
22	Yes	Yes	No	0	0	0
23	Yes	Yes	No	0	0	0
24	Yes	Yes	No	0	0	0
25	Yes	Yes	No	0	0	0
26	Yes	Yes	No	0	0	0
27	Yes	Yes	No	0	0	0
28	Yes	Yes	No	0	0	0
29	Yes	Yes	No	0	0	0
30	Yes	Yes	No	0	0	0
31	Yes	Yes	No	0	0	0
32	Yes	Yes	No	0	0	0
33	Yes	Yes	No	0	0	0
34	Yes	Yes	No	0	0	0
35	Yes	Yes	No	0	0	0
36	Yes	Yes	No	0	0	0
114	Yes	Yes	No	0	0	0
115	Yes	Yes	No	0	0	0
116	Yes	Yes	No	0	0	0
117	Yes	Yes	No	0	0	0
118	Yes	Yes	No	0	0	0

119	Yes	Yes	No	0	0	0
120	Yes	Yes	No	0	0	0
121	Yes	Yes	No	0	0	0
122	Yes	Yes	No	0	0	0
123	Yes	Yes	No	0	0	0
126	Yes	Yes	No	0	0	0
127	Yes	Yes	No	0	0	0
128	Yes	Yes	No	0	0	0
129	Yes	Yes	No	0	0	0
130	Yes	Yes	No	0	0	0
131	Yes	Yes	No	0	0	0
132	Yes	Yes	No	0	0	0
133	Yes	Yes	No	0	0	0
134	Yes	Yes	No	0	0	0
135	Yes	Yes	No	0	0	0
137	Yes	Yes	No	0	0	0
138	Yes	Yes	No	0	0	0
139	Yes	Yes	No	0	0	0
140	Yes	Yes	No	0	0	0
141	Yes	Yes	No	0	0	0
143	Yes	Yes	No	0	0	0
144	Yes	Yes	No	0	0	0
145	Yes	Yes	No	0	0	0
146	Yes	Yes	No	0	0	0
147	Yes	Yes	No	0	0	0
149	Yes	Yes	No	0	0	0
150	Yes	Yes	No	0	0	0
151	Yes	Yes	No	0	0	0
152	Yes	Yes	No	0	0	0
153	Yes	Yes	No	0	0	0
154	Yes	Yes	No	0	0	0
155	Yes	Yes	No	0	0	0
156	Yes	Yes	No	0	0	0
157	Yes	Yes	No	0	0	0
158	Yes	Yes	No	0	0	0
159	Yes	Yes	No	0	0	0
160	Yes	Yes	No	0	0	0
161	Yes	Yes	No	0	0	0
162	Yes	Yes	No	0	0	0
163	Yes	Yes	No	0	0	0
164	Yes	Yes	No	0	0	0
165	Yes	Yes	No	0	0	0
166	Yes	Yes	No	0	0	0
167	Yes	Yes	No	0	0	0
168	Yes	Yes	No	0	0	0
169	Yes	Yes	No	0	0	0

170	Yes	Yes	No	0	0	0
171	Yes	Yes	No	0	0	0
172	Yes	Yes	No	0	0	0
173	Yes	Yes	No	0	0	0
175	Yes	Yes	No	0	0	0
176	Yes	Yes	No	0	0	0
177	Yes	Yes	No	0	0	0
178	Yes	Yes	No	0	0	0
179	Yes	Yes	No	0	0	0
181	Yes	Yes	No	0	0	0
182	Yes	Yes	No	0	0	0
183	Yes	Yes	No	0	0	0
184	Yes	Yes	No	0	0	0
185	Yes	Yes	No	0	0	0
187	Yes	Yes	No	0	0	0
188	Yes	Yes	No	0	0	0
189	Yes	Yes	No	0	0	0
190	Yes	Yes	No	0	0	0
191	Yes	Yes	No	0	0	0
196	Yes	Yes	No	0	0	0
197	Yes	Yes	No	0	0	0
198	Yes	Yes	No	0	0	0
199	Yes	Yes	No	0	0	0
200	Yes	Yes	No	0	0	0
201	Yes	Yes	No	0	0	0
202	Yes	Yes	No	0	0	0
203	Yes	Yes	No	0	0	0
204	Yes	Yes	No	0	0	0
205	Yes	Yes	No	0	0	0
206	Yes	Yes	No	0	0	0
207	Yes	Yes	No	0	0	0
208	Yes	Yes	No	0	0	0
209	Yes	Yes	No	0	0	0
210	Yes	Yes	No	0	0	0
211	Yes	Yes	No	0	0	0
212	Yes	Yes	No	0	0	0
213	Yes	Yes	No	0	0	0
214	Yes	Yes	No	0	0	0
215	Yes	Yes	No	0	0	0

tw	t2b	tfb	Area	Tors Constant	I33	I22	I23	AS2	AS3	S33	S22	Z33
m	m	m	m2	m4	m4	m4	m4	m2	m2	m3	m3	m3
0,01	0,3	0,0175	0,0143	1,53 E-06	0,0003309	7,89E-02	0,00	0,0035	0,00875	0,00189085714285714	0,0005258	0,00208
0,021	0,5	0,04	0,0374	1,58 E-02	0,002816999936	0,0001898	0,00	0,02402	0,0203333333984375	0,00843413154491018	0,00124459016393443	0,00965
0,00659999990463257	0,13	0,01019999990463257	0,0045	1,59 E-07	4,2E-05	3,39E-06	0,00	0,0002295	0,00696666666666666	0,00306933000306933	0,000307909090909091	0,00351
0,012	0,22	0,019	0,0156	1,65 E-06	0,0009208	3,39E-02	0,00	0,0002295	0,00696666666666666	0,00306933000306933	0,000307909090909091	0,00351
			0,306	0,018252	0,0108	0,0108	0,3	0,3	0,3	0,036	0,036	0,054
			0,306	0,018252	0,0108	0,0108	0,3	0,3	0,3	0,036	0,036	0,054

Z22	R33	R22	Conc Col	Conc Beam	Color	TotalWt	TotalMass	Fro mFile	AM od	A2 Mod	A3 Mod	JM od	I2 Mod	I3 Mod	M Mod	W Mod
m3	m	m	Yes/No	Yes/No	Text	KN	KN-s2/m	Yes/No	Unittles	Unittles	Unittles	Unittles	Unittles	Unittles	Unittles	Unittles
0,000802	0,1521172	0,07426563549438	No	No	Red	307.098.454	313.153.443.318.261	Yes	1	1	1	1	1	1	1	1
0,001936	0,2744464499747	0,07123809591897	No	No	Blue	892.423.384.546.618	910.018.553.232.553	Yes	1	1	1	1	1	1	1	1
9,7E-05	0,1123137528754	0,03024950709910	No	No	Green	885.468.277.305.387	902.926.317.933.881	Yes	1	1	1	1	1	1	1	1
0,000486	0,2429519315124	0,04659564192232	No	No	Yellow	864.559.207.799.518	881.605.001.717.098	Yes	1	1	1	1	1	1	1	1
0,054	0,1732050807568	0,17320508075688	No	Yes	2	838.3.778.88.867	3.790.19.90.024	No	6	6	6	6	6	6	1	1
0,054	0,1732050807568	0,17320508075688	Yes	No	08	167.118	0	No	6	6	6	6	6	6	1	1

SectInFile	FileName	GUID	Notes
Text	Text	Text	Text
HE360A	c:\program files\computers and structures\sap2000 17\euro.pro		Imported 16.06.2016 23:44:21 from EURO.PRO
HE650M	c:\program files\computers and structures\sap2000 17\euro.pro		Imported 16.06.2016 23:46:41 from EURO.PRO
IPE270	c:\program files\computers and structures\sap2000 17\euro.pro		Imported 18.06.2019 03:25:42 from EURO.PRO
IPE600	c:\program files\computers and structures\sap2000 17\euro.pro		Imported 16.06.2016 23:46:23 from EURO.PRO
			Added 16.06.2016 23:48:39
			Added 17.06.2016 00:26:52

TABLE: Frame Design Procedures	
Frame	DesignProc
Text	Text
1	From Material
2	From Material
3	From Material
4	From Material
5	From Material
6	From Material

7	From Material
8	From Material
9	From Material
10	From Material
11	From Material
12	From Material
13	From Material
14	From Material
15	From Material
16	From Material
17	From Material
18	From Material
19	From Material
20	From Material
21	From Material
22	From Material
23	From Material
24	From Material
25	From Material
26	From Material
27	From Material
28	From Material
29	From Material
30	From Material
31	From Material
32	From Material
33	From Material
34	From Material
35	From Material
36	From Material
114	From Material
115	From Material
116	From Material
117	From Material
118	From Material
119	From Material
120	From Material
121	From Material
122	From Material
123	From Material
126	From Material
127	From Material
128	From Material
129	From Material
130	From Material
131	From Material

132	From Material
133	From Material
134	From Material
135	From Material
137	From Material
138	From Material
139	From Material
140	From Material
141	From Material
143	From Material
144	From Material
145	From Material
146	From Material
147	From Material
149	From Material
150	From Material
151	From Material
152	From Material
153	From Material
154	From Material
155	From Material
156	From Material
157	From Material
158	From Material
159	From Material
160	From Material
161	From Material
162	From Material
163	From Material
164	From Material
165	From Material
166	From Material
167	From Material
168	From Material
169	From Material
170	From Material
171	From Material
172	From Material
173	From Material
175	From Material
176	From Material
177	From Material
178	From Material
179	From Material
181	From Material
182	From Material

183	From Material
184	From Material
185	From Material
187	From Material
188	From Material
189	From Material
190	From Material
191	From Material
196	From Material
197	From Material
198	From Material
199	From Material
200	From Material
201	From Material
202	From Material
203	From Material
204	From Material
205	From Material
206	From Material
207	From Material
208	From Material
209	From Material
210	From Material
211	From Material
212	From Material
213	From Material
214	From Material
215	From Material

TABLE: Frame Load Transfer Options

Frame	Transfer
Text	Yes/No
1	Yes
2	Yes
3	Yes
4	Yes
5	Yes
6	Yes
7	Yes
8	Yes
9	Yes
10	Yes
11	Yes
12	Yes
13	Yes

14	Yes
15	Yes
16	Yes
17	Yes
18	Yes
19	Yes
20	Yes
21	Yes
22	Yes
23	Yes
24	Yes
25	Yes
26	Yes
27	Yes
28	Yes
29	Yes
30	Yes
31	Yes
32	Yes
33	Yes
34	Yes
35	Yes
36	Yes
114	Yes
115	Yes
116	Yes
117	Yes
118	Yes
119	Yes
120	Yes
121	Yes
122	Yes
123	Yes
126	Yes
127	Yes
128	Yes
129	Yes
130	Yes
131	Yes
132	Yes
133	Yes
134	Yes
135	Yes
137	Yes
138	Yes
139	Yes

140	Yes
141	Yes
143	Yes
144	Yes
145	Yes
146	Yes
147	Yes
149	Yes
150	Yes
151	Yes
152	Yes
153	Yes
154	Yes
155	Yes
156	Yes
157	Yes
158	Yes
159	Yes
160	Yes
161	Yes
162	Yes
163	Yes
164	Yes
165	Yes
166	Yes
167	Yes
168	Yes
169	Yes
170	Yes
171	Yes
172	Yes
173	Yes
175	Yes
176	Yes
177	Yes
178	Yes
179	Yes
181	Yes
182	Yes
183	Yes
184	Yes
185	Yes
187	Yes
188	Yes
189	Yes
190	Yes

191	Yes
196	Yes
197	Yes
198	Yes
199	Yes
200	Yes
201	Yes
202	Yes
203	Yes
204	Yes
205	Yes
206	Yes
207	Yes
208	Yes
209	Yes
210	Yes
211	Yes
212	Yes
213	Yes
214	Yes
215	Yes

TABLE: Frame Output Station Assignments					
Frame	StationType	MinNumSta	MaxStaSpcg	AddAtElmInt	AddAtPtLoad
Text	Text	Unitless	m	Yes/No	Yes/No
1	MinNumSta	3		Yes	Yes
2	MinNumSta	3		Yes	Yes
3	MinNumSta	3		Yes	Yes
4	MinNumSta	3		Yes	Yes
5	MinNumSta	3		Yes	Yes
6	MinNumSta	3		Yes	Yes
7	MinNumSta	3		Yes	Yes
8	MinNumSta	3		Yes	Yes
9	MinNumSta	3		Yes	Yes
10	MinNumSta	3		Yes	Yes
11	MinNumSta	3		Yes	Yes
12	MinNumSta	3		Yes	Yes
13	MinNumSta	3		Yes	Yes
14	MinNumSta	3		Yes	Yes
15	MinNumSta	3		Yes	Yes
16	MinNumSta	3		Yes	Yes
17	MinNumSta	3		Yes	Yes
18	MinNumSta	3		Yes	Yes
19	MinNumSta	3		Yes	Yes
20	MinNumSta	3		Yes	Yes

21	MinNumSta	3		Yes	Yes
22	MinNumSta	3		Yes	Yes
23	MinNumSta	3		Yes	Yes
24	MinNumSta	3		Yes	Yes
25	MinNumSta	3		Yes	Yes
26	MinNumSta	3		Yes	Yes
27	MinNumSta	3		Yes	Yes
28	MinNumSta	3		Yes	Yes
29	MinNumSta	3		Yes	Yes
30	MinNumSta	3		Yes	Yes
31	MinNumSta	3		Yes	Yes
32	MinNumSta	3		Yes	Yes
33	MinNumSta	3		Yes	Yes
34	MinNumSta	3		Yes	Yes
35	MinNumSta	3		Yes	Yes
36	MinNumSta	3		Yes	Yes
114	MinNumSta	3		Yes	Yes
115	MinNumSta	3		Yes	Yes
116	MinNumSta	3		Yes	Yes
117	MinNumSta	3		Yes	Yes
118	MinNumSta	3		Yes	Yes
119	MinNumSta	3		Yes	Yes
120	MinNumSta	3		Yes	Yes
121	MinNumSta	3		Yes	Yes
122	MinNumSta	3		Yes	Yes
123	MinNumSta	3		Yes	Yes
126	MinNumSta	3		Yes	Yes
127	MinNumSta	3		Yes	Yes
128	MinNumSta	3		Yes	Yes
129	MinNumSta	3		Yes	Yes
130	MinNumSta	3		Yes	Yes
131	MinNumSta	3		Yes	Yes
132	MinNumSta	3		Yes	Yes
133	MinNumSta	3		Yes	Yes
134	MinNumSta	3		Yes	Yes
135	MinNumSta	3		Yes	Yes
137	MaxStaSpcg		0,5	Yes	Yes
138	MaxStaSpcg		0,5	Yes	Yes
139	MaxStaSpcg		0,5	Yes	Yes
140	MaxStaSpcg		0,5	Yes	Yes
141	MaxStaSpcg		0,5	Yes	Yes
143	MaxStaSpcg		0,5	Yes	Yes
144	MaxStaSpcg		0,5	Yes	Yes
145	MaxStaSpcg		0,5	Yes	Yes
146	MaxStaSpcg		0,5	Yes	Yes
147	MaxStaSpcg		0,5	Yes	Yes

149	MaxStaSpcg		0,5	Yes	Yes
150	MaxStaSpcg		0,5	Yes	Yes
151	MaxStaSpcg		0,5	Yes	Yes
152	MaxStaSpcg		0,5	Yes	Yes
153	MaxStaSpcg		0,5	Yes	Yes
154	MinNumSta	3		Yes	Yes
155	MinNumSta	3		Yes	Yes
156	MinNumSta	3		Yes	Yes
157	MinNumSta	3		Yes	Yes
158	MinNumSta	3		Yes	Yes
159	MinNumSta	3		Yes	Yes
160	MinNumSta	3		Yes	Yes
161	MinNumSta	3		Yes	Yes
162	MinNumSta	3		Yes	Yes
163	MinNumSta	3		Yes	Yes
164	MinNumSta	3		Yes	Yes
165	MinNumSta	3		Yes	Yes
166	MinNumSta	3		Yes	Yes
167	MinNumSta	3		Yes	Yes
168	MinNumSta	3		Yes	Yes
169	MinNumSta	3		Yes	Yes
170	MinNumSta	3		Yes	Yes
171	MinNumSta	3		Yes	Yes
172	MinNumSta	3		Yes	Yes
173	MinNumSta	3		Yes	Yes
175	MaxStaSpcg		0,5	Yes	Yes
176	MaxStaSpcg		0,5	Yes	Yes
177	MaxStaSpcg		0,5	Yes	Yes
178	MaxStaSpcg		0,5	Yes	Yes
179	MaxStaSpcg		0,5	Yes	Yes
181	MaxStaSpcg		0,5	Yes	Yes
182	MaxStaSpcg		0,5	Yes	Yes
183	MaxStaSpcg		0,5	Yes	Yes
184	MaxStaSpcg		0,5	Yes	Yes
185	MaxStaSpcg		0,5	Yes	Yes
187	MaxStaSpcg		0,5	Yes	Yes
188	MaxStaSpcg		0,5	Yes	Yes
189	MaxStaSpcg		0,5	Yes	Yes
190	MaxStaSpcg		0,5	Yes	Yes
191	MaxStaSpcg		0,5	Yes	Yes
196	MaxStaSpcg		0,5	Yes	Yes
197	MaxStaSpcg		0,5	Yes	Yes
198	MaxStaSpcg		0,5	Yes	Yes
199	MaxStaSpcg		0,5	Yes	Yes
200	MaxStaSpcg		0,5	Yes	Yes
201	MaxStaSpcg		0,5	Yes	Yes

202	MaxStaSpcg		0,5	Yes	Yes
203	MaxStaSpcg		0,5	Yes	Yes
204	MaxStaSpcg		0,5	Yes	Yes
205	MaxStaSpcg		0,5	Yes	Yes
206	MaxStaSpcg		0,5	Yes	Yes
207	MaxStaSpcg		0,5	Yes	Yes
208	MaxStaSpcg		0,5	Yes	Yes
209	MaxStaSpcg		0,5	Yes	Yes
210	MaxStaSpcg		0,5	Yes	Yes
211	MaxStaSpcg		0,5	Yes	Yes
212	MaxStaSpcg		0,5	Yes	Yes
213	MaxStaSpcg		0,5	Yes	Yes
214	MaxStaSpcg		0,5	Yes	Yes
215	MaxStaSpcg		0,5	Yes	Yes

TABLE: Frame Section Properties 02 - Concrete Column

SectionName	RebarMatL	RebarMatC	ReinfConfig	LatReinf	Concrete	NumBars3Di	NumBars2Di	BarSizeL	BarSizeC	SpacingC	NumBars2	NumBars3	ReinfType
Text	Text	Text	Text	Text	m	Unitless	Unitless	Text	Text	m	Unitless	Unitless	Text
Kolon	S420	S420	Rectangular	Ties	0,04	6	6	N24	N24	0,15	6	6	Design

TABLE: Frame Section Properties 03 - Concrete Beam

SectionName	RebarMatL	RebarMatC	TopCover	BotCover	TopLeftArea	TopRightArea	BotLeftArea	BotRightArea
Text	Text	Text	m	m	m2	m2	m2	m2
Kiris	S420	S420	0,06	0,06	0	0	0	0

TABLE: Frame Section Properties 13 - Time Dependent

SectionName	TypeSize	AutoValSize	AutoSFSize	UserValSize
Text	Text	m	Unitless	m
HE360A	Auto	0	0	
HE650M	Auto	0	0	
IPE270	Auto	0,25	1	
IPE600	Auto	0	0	
Kiris	Auto	0,3	1	
Kolon	Auto	0,3	1	

TABLE: Frame Section Assignments					
Frame	SectionType	AutoSelect	AnalSect	DesignSect	MatProp
Text	Text	Text	Text	Text	Text
1	I/Wide Flange	N.A.	IPE270	IPE270	Default
2	I/Wide Flange	N.A.	IPE270	IPE270	Default
3	I/Wide Flange	N.A.	IPE270	IPE270	Default
4	I/Wide Flange	N.A.	IPE270	IPE270	Default
5	I/Wide Flange	N.A.	IPE270	IPE270	Default
6	I/Wide Flange	N.A.	IPE270	IPE270	Default
7	I/Wide Flange	N.A.	IPE270	IPE270	Default
8	I/Wide Flange	N.A.	IPE270	IPE270	Default
9	I/Wide Flange	N.A.	IPE270	IPE270	Default
10	I/Wide Flange	N.A.	IPE270	IPE270	Default
11	I/Wide Flange	N.A.	IPE270	IPE270	Default
12	I/Wide Flange	N.A.	IPE270	IPE270	Default
13	I/Wide Flange	N.A.	IPE270	IPE270	Default
14	I/Wide Flange	N.A.	IPE270	IPE270	Default
15	I/Wide Flange	N.A.	IPE270	IPE270	Default
16	I/Wide Flange	N.A.	IPE270	IPE270	Default
17	I/Wide Flange	N.A.	IPE270	IPE270	Default
18	I/Wide Flange	N.A.	IPE270	IPE270	Default
19	I/Wide Flange	N.A.	IPE270	IPE270	Default
20	I/Wide Flange	N.A.	IPE270	IPE270	Default
21	I/Wide Flange	N.A.	IPE270	IPE270	Default
22	I/Wide Flange	N.A.	IPE270	IPE270	Default
23	I/Wide Flange	N.A.	IPE270	IPE270	Default
24	I/Wide Flange	N.A.	IPE270	IPE270	Default
25	I/Wide Flange	N.A.	IPE270	IPE270	Default
26	I/Wide Flange	N.A.	IPE270	IPE270	Default
27	I/Wide Flange	N.A.	IPE270	IPE270	Default
28	I/Wide Flange	N.A.	IPE270	IPE270	Default
29	I/Wide Flange	N.A.	IPE270	IPE270	Default
30	I/Wide Flange	N.A.	IPE270	IPE270	Default
31	I/Wide Flange	N.A.	IPE270	IPE270	Default
32	I/Wide Flange	N.A.	IPE270	IPE270	Default
33	I/Wide Flange	N.A.	IPE270	IPE270	Default
34	I/Wide Flange	N.A.	IPE270	IPE270	Default
35	I/Wide Flange	N.A.	IPE270	IPE270	Default
36	I/Wide Flange	N.A.	IPE270	IPE270	Default
114	I/Wide Flange	N.A.	HE360A	HE360A	Default
115	I/Wide Flange	N.A.	HE360A	HE360A	Default
116	I/Wide Flange	N.A.	HE360A	HE360A	Default
117	I/Wide Flange	N.A.	HE360A	HE360A	Default
118	I/Wide Flange	N.A.	HE360A	HE360A	Default
119	I/Wide Flange	N.A.	HE360A	HE360A	Default
120	I/Wide Flange	N.A.	HE360A	HE360A	Default

121	I/Wide Flange	N.A.	HE360A	HE360A	Default
122	I/Wide Flange	N.A.	HE360A	HE360A	Default
123	I/Wide Flange	N.A.	HE360A	HE360A	Default
126	I/Wide Flange	N.A.	HE650M	HE650M	Default
127	I/Wide Flange	N.A.	HE650M	HE650M	Default
128	I/Wide Flange	N.A.	HE650M	HE650M	Default
129	I/Wide Flange	N.A.	HE650M	HE650M	Default
130	I/Wide Flange	N.A.	HE650M	HE650M	Default
131	I/Wide Flange	N.A.	HE650M	HE650M	Default
132	I/Wide Flange	N.A.	HE650M	HE650M	Default
133	I/Wide Flange	N.A.	HE650M	HE650M	Default
134	I/Wide Flange	N.A.	HE650M	HE650M	Default
135	I/Wide Flange	N.A.	HE650M	HE650M	Default
137	I/Wide Flange	N.A.	IPE600	IPE600	Default
138	I/Wide Flange	N.A.	IPE600	IPE600	Default
139	I/Wide Flange	N.A.	IPE600	IPE600	Default
140	I/Wide Flange	N.A.	IPE600	IPE600	Default
141	I/Wide Flange	N.A.	IPE600	IPE600	Default
143	I/Wide Flange	N.A.	IPE600	IPE600	Default
144	I/Wide Flange	N.A.	IPE600	IPE600	Default
145	I/Wide Flange	N.A.	IPE600	IPE600	Default
146	I/Wide Flange	N.A.	IPE600	IPE600	Default
147	I/Wide Flange	N.A.	IPE600	IPE600	Default
149	Rectangular	N.A.	Kiris	Kiris	Default
150	Rectangular	N.A.	Kiris	Kiris	Default
151	Rectangular	N.A.	Kiris	Kiris	Default
152	Rectangular	N.A.	Kiris	Kiris	Default
153	Rectangular	N.A.	Kiris	Kiris	Default
154	I/Wide Flange	N.A.	HE360A	HE360A	Default
155	I/Wide Flange	N.A.	HE360A	HE360A	Default
156	I/Wide Flange	N.A.	HE360A	HE360A	Default
157	I/Wide Flange	N.A.	HE360A	HE360A	Default
158	I/Wide Flange	N.A.	HE360A	HE360A	Default
159	I/Wide Flange	N.A.	HE360A	HE360A	Default
160	I/Wide Flange	N.A.	HE360A	HE360A	Default
161	I/Wide Flange	N.A.	HE360A	HE360A	Default
162	I/Wide Flange	N.A.	HE360A	HE360A	Default
163	I/Wide Flange	N.A.	HE360A	HE360A	Default
164	I/Wide Flange	N.A.	HE360A	HE360A	Default
165	I/Wide Flange	N.A.	HE360A	HE360A	Default
166	I/Wide Flange	N.A.	HE360A	HE360A	Default
167	I/Wide Flange	N.A.	HE360A	HE360A	Default
168	I/Wide Flange	N.A.	HE360A	HE360A	Default
169	I/Wide Flange	N.A.	HE360A	HE360A	Default
170	I/Wide Flange	N.A.	HE360A	HE360A	Default
171	I/Wide Flange	N.A.	HE360A	HE360A	Default

172	I/Wide Flange	N.A.	HE360A	HE360A	Default
173	I/Wide Flange	N.A.	HE360A	HE360A	Default
175	I/Wide Flange	N.A.	HE360A	HE360A	Default
176	I/Wide Flange	N.A.	HE360A	HE360A	Default
177	I/Wide Flange	N.A.	HE360A	HE360A	Default
178	I/Wide Flange	N.A.	HE360A	HE360A	Default
179	I/Wide Flange	N.A.	HE360A	HE360A	Default
181	I/Wide Flange	N.A.	HE360A	HE360A	Default
182	I/Wide Flange	N.A.	HE360A	HE360A	Default
183	I/Wide Flange	N.A.	HE360A	HE360A	Default
184	I/Wide Flange	N.A.	HE360A	HE360A	Default
185	I/Wide Flange	N.A.	HE360A	HE360A	Default
187	I/Wide Flange	N.A.	HE360A	HE360A	Default
188	I/Wide Flange	N.A.	HE360A	HE360A	Default
189	I/Wide Flange	N.A.	HE360A	HE360A	Default
190	I/Wide Flange	N.A.	HE360A	HE360A	Default
191	I/Wide Flange	N.A.	HE360A	HE360A	Default
196	I/Wide Flange	N.A.	IPE600	IPE600	Default
197	I/Wide Flange	N.A.	IPE600	IPE600	Default
198	Rectangular	N.A.	Kiris	Kiris	Default
199	Rectangular	N.A.	Kiris	Kiris	Default
200	I/Wide Flange	N.A.	HE360A	HE360A	Default
201	I/Wide Flange	N.A.	HE360A	HE360A	Default
202	I/Wide Flange	N.A.	HE360A	HE360A	Default
203	I/Wide Flange	N.A.	HE360A	HE360A	Default
204	I/Wide Flange	N.A.	HE360A	HE360A	Default
205	I/Wide Flange	N.A.	HE360A	HE360A	Default
206	I/Wide Flange	N.A.	HE360A	HE360A	Default
207	I/Wide Flange	N.A.	HE360A	HE360A	Default
208	I/Wide Flange	N.A.	HE360A	HE360A	Default
209	I/Wide Flange	N.A.	HE360A	HE360A	Default
210	I/Wide Flange	N.A.	HE360A	HE360A	Default
211	I/Wide Flange	N.A.	HE360A	HE360A	Default
212	I/Wide Flange	N.A.	HE360A	HE360A	Default
213	I/Wide Flange	N.A.	HE360A	HE360A	Default
214	I/Wide Flange	N.A.	HE360A	HE360A	Default
215	I/Wide Flange	N.A.	HE360A	HE360A	Default

TABLE: Function - Plot Functions				
PlotFunc	Type	DistType	Component	Mode
Text	Text	Text	Text	Text
Input Energy	Energy		Input	All

TABLE: Function - Power Spectral Density - User

Name	Frequency	Value
Text	Cyc/sec	Unitless
UNIFPSD	0	1
UNIFPSD	1	1

TABLE: Function - Response Spectrum - User

Name	Period	Accel	FuncDamp
Text	Sec	Unitless	Unitless
UNIFRS	0	1	0,05
UNIFRS	1	1	

TABLE: Function - Steady State - User

Name	Frequency	Value
Text	Cyc/sec	Unitless
UNIFSS	0	1
UNIFSS	1	1

TABLE: Function - Time History - User

Name	Time	Value
Text	Sec	Unitless
RAMPTH	0	0
RAMPTH	1	1
RAMPTH	4	1
UNIFTH	0	1
UNIFTH	1	1

TABLE: Grid Lines

CoordSys	AxisDir	GridID	XRYZCoord	LineType	LineColor	Visible	BubbleLoc	AllVisible	BubbleSize
Text	Text	Text	m	Text	Text	Yes/No	Text	Yes/No	m
GLOBAL	X	x	0	Primary	Gray8Dark	Yes	End	Yes	1,25
GLOBAL	X	x6	6	Primary	Gray8Dark	Yes	End		
GLOBAL	X	x12	12	Primary	Gray8Dark	Yes	End		
GLOBAL	X	x18	18	Primary	Gray8Dark	Yes	End		
GLOBAL	Y	y	0	Primary	Gray8Dark	Yes	Start		

GLOBA L	Y	y2	6	Primar y	Gray8Da rk	Yes	Start
GLOBA L	Z	z	0	Primar y	Gray8Da rk	Yes	End
GLOBA L	Z	z2	3,1	Primar y	Gray8Da rk	Yes	End
GLOBA L	Z	z2	6,2	Primar y	Gray8Da rk	Yes	End
GLOBA L	Z	z2	9,3	Primar y	Gray8Da rk	Yes	End
GLOBA L	Z	z2	12,4	Primar y	Gray8Da rk	Yes	End
GLOBA L	Z	z2	15,5	Primar y	Gray8Da rk	Yes	End

**TABLE: Groups
1 - Definitions**

Group Name	Selecti on	Section Cut	Stee l	Concr ete	Alum inum	Cold Formed	Stag e	Bridg e	Auto Seis mic	Aut oWind	SelD esSt eel	SelD esAl um	SelD esC old	Mass Weig ht	C ol or
Text	Yes /No	Yes/ No	Ye s/ No	Yes /No	Yes/ No	Yes/ No	Ye s/ No	Ye s/ No	Yes/ No	Yes/ No	Yes/ No	Yes/ No	Yes/ No	Yes/ No	T e x t
ALL	Yes	Yes	Ye s	Yes	Yes	Yes	Ye s	Ye s	No	No	No	No	No	Yes	R e d

TABLE: Groups 3 - Masses and Weights

Group Name	SelfMass	SelfWeight	TotalMassX	TotalMassY	TotalMassZ
Text	KN-s2/m	KN	KN-s2/m	KN-s2/m	KN-s2/m
ALL	109.088.578.5 99.271	106.978.345.6 23.377	109.088.578.5 99.271	109.088.578.5 99.271	109.088.578.5 99.271

**TABLE: Joint
Coordinates**

Joint	Coord Sys	Coord Type	Xo rR	Y	Z	Speci alJt	Glob alX	Glob alY	Glob alZ	GUID
Text	Text	Text	m	m	m	Yes/N o	m	m	m	Text
49	GLOB AL	Cartesi an	0	0	0	No	0	0	0	217e321d-9844-413c-9cf0- 0256123d706b
50	GLOB AL	Cartesi an	0	0	1	No	0	0	3,1	217e321d-9844-413c-9cf0- 0256123d706b
51	GLOB AL	Cartesi an	0	0	2	No	0	0	6,2	217e321d-9844-413c-9cf0- 0256123d706b
52	GLOB	Cartesi	0	0	9,	No	0	0	9,3	217e321d-9844-413c-9cf0-

	AL	an			3					0256123d706b
	GLOB	Cartesi			12					217e321d-9844-413c-9cf0-
53	AL	an	0	0	,4	No	0	0	12,4	0256123d706b
	GLOB	Cartesi			15					217e321d-9844-413c-9cf0-
54	AL	an	0	0	,5	No	0	0	15,5	0256123d706b
	GLOB	Cartesi								217e321d-9844-413c-9cf0-
55	AL	an	18	0	0	No	18	0	0	0256123d706b
	GLOB	Cartesi			3,					217e321d-9844-413c-9cf0-
56	AL	an	18	0	1	No	18	0	3,1	0256123d706b
	GLOB	Cartesi			6,					217e321d-9844-413c-9cf0-
57	AL	an	18	0	2	No	18	0	6,2	0256123d706b
	GLOB	Cartesi			9,					217e321d-9844-413c-9cf0-
58	AL	an	18	0	3	No	18	0	9,3	0256123d706b
	GLOB	Cartesi			12					217e321d-9844-413c-9cf0-
59	AL	an	18	0	,4	No	18	0	12,4	0256123d706b
	GLOB	Cartesi			15					217e321d-9844-413c-9cf0-
60	AL	an	18	0	,5	No	18	0	15,5	0256123d706b
	GLOB	Cartesi								217e321d-9844-413c-9cf0-
64	AL	an	6	0	0	No	6	0	0	0256123d706b
	GLOB	Cartesi			3,					217e321d-9844-413c-9cf0-
65	AL	an	6	0	1	No	6	0	3,1	0256123d706b
	GLOB	Cartesi			6,					217e321d-9844-413c-9cf0-
66	AL	an	6	0	2	No	6	0	6,2	0256123d706b
	GLOB	Cartesi			9,					217e321d-9844-413c-9cf0-
67	AL	an	6	0	3	No	6	0	9,3	0256123d706b
	GLOB	Cartesi			12					217e321d-9844-413c-9cf0-
68	AL	an	6	0	,4	No	6	0	12,4	0256123d706b
	GLOB	Cartesi			15					217e321d-9844-413c-9cf0-
69	AL	an	6	0	,5	No	6	0	15,5	0256123d706b
	GLOB	Cartesi								217e321d-9844-413c-9cf0-
70	AL	an	12	0	0	No	12	0	0	0256123d706b
	GLOB	Cartesi			3,					217e321d-9844-413c-9cf0-
71	AL	an	12	0	1	No	12	0	3,1	0256123d706b
	GLOB	Cartesi			6,					217e321d-9844-413c-9cf0-
72	AL	an	12	0	2	No	12	0	6,2	0256123d706b
	GLOB	Cartesi			9,					217e321d-9844-413c-9cf0-
73	AL	an	12	0	3	No	12	0	9,3	0256123d706b
	GLOB	Cartesi			12					217e321d-9844-413c-9cf0-
74	AL	an	12	0	,4	No	12	0	12,4	0256123d706b
	GLOB	Cartesi			15					217e321d-9844-413c-9cf0-
75	AL	an	12	0	,5	No	12	0	15,5	0256123d706b
	GLOB	Cartesi								217e321d-9844-413c-9cf0-
76	AL	an	0	6	0	No	0	6	0	0256123d706b
	GLOB	Cartesi			3,					217e321d-9844-413c-9cf0-
77	AL	an	0	6	1	No	0	6	3,1	0256123d706b
	GLOB	Cartesi			6,					217e321d-9844-413c-9cf0-
78	AL	an	0	6	2	No	0	6	6,2	0256123d706b
	GLOB	Cartesi			9,					217e321d-9844-413c-9cf0-
79	AL	an	0	6	3	No	0	6	9,3	0256123d706b
	GLOB	Cartesi			12					217e321d-9844-413c-9cf0-
80	AL	an	0	6	,4	No	0	6	12,4	0256123d706b

81	GLOB	Cartesi		15						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	0	6	,5	No	0	6	15,5	
82	GLOB	Cartesi								217e321d-9844-413c-9cf0-0256123d706b
	AL	an	18	6	0	No	18	6	0	
83	GLOB	Cartesi		3,						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	18	6	1	No	18	6	3,1	
84	GLOB	Cartesi		6,						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	18	6	2	No	18	6	6,2	
85	GLOB	Cartesi		9,						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	18	6	3	No	18	6	9,3	
86	GLOB	Cartesi		12						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	18	6	,4	No	18	6	12,4	
87	GLOB	Cartesi		15						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	18	6	,5	No	18	6	15,5	
88	GLOB	Cartesi								217e321d-9844-413c-9cf0-0256123d706b
	AL	an	6	6	0	No	6	6	0	
89	GLOB	Cartesi		3,						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	6	6	1	No	6	6	3,1	
90	GLOB	Cartesi		6,						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	6	6	2	No	6	6	6,2	
91	GLOB	Cartesi		9,						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	6	6	3	No	6	6	9,3	
92	GLOB	Cartesi		12						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	6	6	,4	No	6	6	12,4	
93	GLOB	Cartesi		15						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	6	6	,5	No	6	6	15,5	
94	GLOB	Cartesi								217e321d-9844-413c-9cf0-0256123d706b
	AL	an	12	6	0	No	12	6	0	
95	GLOB	Cartesi		3,						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	12	6	1	No	12	6	3,1	
96	GLOB	Cartesi		6,						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	12	6	2	No	12	6	6,2	
97	GLOB	Cartesi		9,						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	12	6	3	No	12	6	9,3	
98	GLOB	Cartesi		12						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	12	6	,4	No	12	6	12,4	
99	GLOB	Cartesi		15						217e321d-9844-413c-9cf0-0256123d706b
	AL	an	12	6	,5	No	12	6	15,5	

TABLE: Joint Pattern Definitions						
Pattern						
Text						

Default

TABLE: Joint Restraint Assignments						
Joint	U1	U2	U3	R1	R2	R3
Text	Yes/No	Yes/No	Yes/No	Yes/No	Yes/No	Yes/No
49	Yes	Yes	Yes	Yes	Yes	Yes

55	Yes	Yes	Yes	Yes	Yes	Yes	Yes
64	Yes	Yes	Yes	Yes	Yes	Yes	Yes
70	Yes	Yes	Yes	Yes	Yes	Yes	Yes
76	Yes	Yes	Yes	Yes	Yes	Yes	Yes
82	Yes	Yes	Yes	Yes	Yes	Yes	Yes
88	Yes	Yes	Yes	Yes	Yes	Yes	Yes
94	Yes	Yes	Yes	Yes	Yes	Yes	Yes

TABLE: Load Case Definitions

Case	Type	Initial Cond	Modal Case	Base Case	Mass Source	Dest Type	Design Type	Design Act Opt	Design Act	Auto Type	Run Case	Case Status	GUID	Notes
Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	Yes/No	Text	Text	Text
DEAD	LinS	Zero				Prog Det	DEA	Prog Det	Non-Composite	Non	Yes	Not Run		
MODAL	LinS	Zero				Prog Det	OTHER	Prog Det	Other	Non	Yes	Not Run		
H	LinS	Zero				Prog Det	LIVE	Prog Det	Short-Term Composite	Non	Yes	Not Run		
Kar	LinS	Zero				Prog Det	WIN D	Prog Det	Short-Term Composite	Non	Yes	Not Run		
E	LinS	Zero				Prog Det	QUAKE	Prog Det	Short-Term Composite	Non	Yes	Not Run		

TABLE: Load Pattern Definitions

Load Pattern	Design Type	SelfWtMult	AutoLoad	GUID	Notes
Text	Text	Unitless	Text	Text	Text
D	DEAD	1		1c5eae18-92cc-4dfb-a23c-b979cde32fbc	
H	LIVE	0		bfa1b01a-d63d-4824-993b-eef9754c995d	
S	WIND	0	TS 498-97	aa79166b-3027-4571-bbfa-dbfac3cc8c72	
E	QUAKE	0	TSC 2007	bfe2b299-fde4-4a64-9a88-	

TABLE: Mass Source				
MassSource	Elements	Masses	Loads	IsDefault
Text	Yes/No	Yes/No	Yes/No	Yes/No
MSSSRC1	Yes	Yes	No	Yes

TABLE: Material List 1 - By Object Type			
ObjectType	Material	TotalWeight	NumPieces
Text	Text	KN	Unitless
Frame	C40/50	3.778.883.790.192	7
Frame	Steel	571.343.722.235.607	119
Area	C40/50	960.016.429.555.919	
Area	Cephe	148.438.165.523.733	

TABLE: Material List 2 - By Section Property				
Section	ObjectType	NumPieces	TotalLength	TotalWeight
Text	Text	Unitless	m	KN
HE360A	Frame	61	279	307.098.635.270.454
IPE600	Frame	12	72	864.559.207.799.518
HE650M	Frame	10	31	892.423.384.546.618
Kiris	Frame	7	42	3.778.883.790.192
IPE270	Frame	36	250.624.010.638.956	885.468.277.305.387
DOSEME	Area			960.016.429.555.919
Cephe	Area			148.438.165.523.733

TABLE: Material Properties 01 - General						
Material	Type	SymType	TempDepend	Color	GUID	Notes
Text	Text	Text	Yes/No	Text	Text	Text
C40/50	Concrete	Isotropic	No	Green		Europe EN 1992-1-1 per EN 206-1 C40/50 added 16.06.2016 20:10:57
Ceph	Aluminum	Isotropic	No	Green		Aluminum added 18.06.2019 03:15:06
S420	Rebar	Uniaxial	No	Green		Rebar added 16.06.2016 23:38:11
Steel	Steel	Isotropic	No	Green		Europe EN 1993-1-1 per EN 10025-2 S450 added 16.06.2016 23:40:25

TABLE: Material Properties 02 - Basic Mechanical Properties

Material	UnitWeight	UnitMass	E1	G12	U12	A1
Text	KN/m3	KN-s2/m4	KN/m2	KN/m2	Unitless	1/C
C40/50	2.499.261.766	25.485.377	35000000	145.833.333.33	0,2	1,00E-05
Cephe	266.018.217.78	271.447.161.00	696.370.546.84	261.793.438.66	0,33	2,36E+08
S420	4.467	4.558	1.094	2.065		1,2E-05
Steel	0	0	200000000	807.692.307.69	0,3	1,17E-05
	769.728.639.42	784.904.737.99		2.308		
	2.648	5.992	210000000			

TABLE: Material Properties 03a - Steel Data

Material	Fy	Fu	EffFy	EffFu	SSCurve Opt	SSHysType	SHard	SMax	SRup	FinalSlope
Text	KN/m2	KN/m2	KN/m2	KN/m2	Text	Text	Unitless	Unitless	Unitless	Unitless
Steel	4400	5500	4840	6050	Simple	Kinematic	0,015	0,11	0,17	-0,1

TABLE: Material Properties 03b - Concrete Data

Material	Fc	LtWtConc	SSCurve Opt	SSHysType	SFc	SCap	FinalSlope	FAnge	DAnge
Text	KN/m2	Yes/No	Text	Text	Unitless	Unitless	Unitless	Degrees	Degrees
C40/50	4000	No	Mander	Takeda	0,002285	0,005	-0,1	0	0

TABLE: Material Properties 03c - Aluminum Data

Material	AlumType	Alloy	Fcy	Fty	Ftu	Fsu	SSHysType
Text	Text	Text	KN/m2	KN/m2	KN/m2	KN/m2	Text
Ceph	Wrought	6061-T6	241.316.526.	241.316.526.	2.620.007.9	165.474.189.	Kinematic
			133.052	133.052	98.016	348.379	

TABLE: Material Properties 03e - Rebar Data

Material	Fy	Fu	EffFy	EffFu	SSCurve Opt	SSHysType	SHard	SCap	FinalSlope	UseCT Def
Text	KN/m	KN/m	KN/m	KN/m	Text	Text	Unitless	Unitless	Unitless	Yes/No
S420	4200	4200	4200	4200	Simple	Kinematic	0,01	0,09	-0,1	No

TABLE: Material Properties 04 - User Stress-Strain Curves

Material	Point	Strain	Stress	PointID
Text	Text	Unitless	KN/m2	Text
Cephe	1	-0,00346534653465347	-241.316.526.133.052	
Cephe	2	0	0	A
Cephe	3	0,00346534653465347	241.316.526.133.052	

TABLE: Material Properties 06 - Damping Parameters

Material	ModalRatio	VisMass	VisStiff	HysMass	HysStiff
Text	Unitless	1/Sec	Sec	1/Sec2	Unitless
C40/50	0	0	0	0	0
Cephe	0	0	0	0	0
S420	0	0	0	0	0
Steel	0	0	0	0	0

TABLE: Overwrites - Concrete Design - ACI 318-14

Frame	DesignSector	FrameType	RLLF	XL Major	XL Minor	XK Major	XK Minor	Cm Major	Cm Minor	Dns Major	Dns Minor	Ds Major	Ds Minor
Text	Text	Text	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless
14	Program Determined	Program Determined	0	0	0								
15	Program Determined	Program Determined	0	0	0								
15	Program Determined	Program Determined	0	0	0								
15	Program Determined	Program Determined	0	0	0								

	ed	ed			
	Program	Program			
15	Determin	Determin	0	0	0
3	ed	ed			
	Program	Program			
19	Determin	Determin	0	0	0
8	ed	ed			
	Program	Program			
19	Determin	Determin	0	0	0
9	ed	ed			

TABLE: Preferences - Aluminum Design - AA-ASD 2000

THDesign	FrameType	SRatioLimit	MaxIter	LatFact	UseLatFact	Bridge
Text	Text	Unitless	Unitless	Unitless	Yes/No	Yes/No
Envelopes	Moment Frame	1	1	133.333.333.333.33	No	No

TABLE: Preferences - Cold Formed Design - AISI-ASD96

THDesign	FrameType	SRatioLimit	MaxIter	OmegaBS	OmegaBUS	OmegaBLTB	OmegaBS	OmegaVNS	OmegaT	OmegaC
Text	Text	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless
Envelopes	Braced Frame	1	1	1,67	1,67	1,67	1,67	1,5	1,67	1,8

TABLE: Preferences - Concrete Design - ACI 318-14

THDesign	NumCurves	NumPoints	MinEccen	PatLF	UFLimit	SeisCat	PhiT	PhiC Tied	PhiCSpiral	PhiV	PhiVSeismic	PhiVJoint
Text	Unitless	Unitless	Yes/No	Unitless	Unitless	Text	Unitless	Unitless	Unitless	Unitless	Unitless	Unitless
Envelopes	24	11	Yes	0,75	0,95	D	0,9	0,65	0,75	0,75	0,6	0,85

TABLE: Preferences - Steel Design - AISC 360-10

THDesign	FrameType	PatLLF	SRatioLimit	MaxIter	SDC	SeisCode	SeisLoad	ImpFactor	SystemRho
Text	Text	Unitless	Unitless	Unitless	Text	Yes/No	Yes/No	Unitless	Unitless
Envelopes	SMF	0,75	0,95	1	D	Yes	Yes	1	1

Syste mSds	Syst em R	Syst emC d	Om ega O	Prov ision	AMeth od	SOMeth od	SRM etho d	NLC oef f	Phi B	Phi C	Phi TY	Phi TF	Phi V
Unitl ess	Unit less	Unitl ess	Unit less	Text	Text	Text	Text	Uni tles s	Uni tles s	Uni tles s	Uni tles s	Uni tles s	Uni tles s
0,5	8	5,5	3	LRFD	Direct Analysis	General 2nd Order	Tau-b Fixed	0,0 02	0,9	0,9	0,9	0,7 5	0,9

PhiVRol ledl	PhiV T	PlugW eld	HSSWel ding	HSSRed uceT	Check Defl	DLRa t	SDLAndL LRat	LLRa t	Total Rat	NetR at
Unitless	Unitl ess	Yes/N o	Text	Yes/No	Yes/N o	Unitl ess	Unitless	Unitl ess	Unitl ess	Unitl ess
1	0,9	Yes	ERW	No	No	120	120	360	240	240

TABLE: Preferences - Dimensional

Merg eTol	Fine Grid	Nud ge	Selec tTol	Snap Tol	SLineT hick	PLineT hick	MaxF ont	MinF ont	AutoZ oom	Shrink Fact	TextFil eLen
m	m	m	Unitl ess	Unitl ess	Unitles s	Unitles s	Unitl ess	Unitl ess	Unitless s	Unitless s	Unitless s
0,001	0,25	0,2 5	3	12	2	4	8	3	10	70	240

TABLE: Program Control

Progra mName	Ver sion	Pro gLe vel	LicenseNu m	Lice nse OS	Lice nse SC	Lice nse HT	Curr Unit s	Steel Code	Conc Cod e	Alum Code	Cold Cod e	Rege nHin ge
Text	Text	Text	Text	Yes/ No	Yes/ No	Yes/ No	Text	Text	Text	Text	Text	Yes/ No
SAP20 00	17. 3.0	Ulti mat e	3010*1HKK 4XU8HYNE GKE	Yes	Yes	No	KN, m, C	AISC 360- 10	ACI 318- 14	AA- 2000	AISI- ASD 96	Yes

TABLE: Rebar Sizes

RebarID	Area	Diameter
Text	m2	m
#2	3,23E-01	0,00635

#3	7,10E+09	0,009525
#4	0,000129032001922727	0,0127
#5	0,000199999601538181	0,015875
#6	0,000283870398461819	0,01905
#7	0,000387096015381813	0,022225
#8	0,000509676413843632	0,0254
#9	0,00064516	0,0286512005329132
#10	0,00081935318769455	0,0322579995155334
#11	0,00100644956308365	0,0358139991521835
#14	0,00145161	0,0430021989583969
#18	0,00258064	0,0573277992248535
10M	0,000100000002655387	0,011300000667572
15M	0,000200000005310774	0,0159999999403954
20M	0,0003000000017579794	0,0194999997854233
25M	0,000500000022890568	0,0252000016093254
30M	0,000700000008974075	0,0299000008821487
35M	0,00100000004578114	0,0357000011444092
45M	0,00150000010712624	0,0437000003576279
55M	0,00250000007599831	0,0564000033855438
6d	2,83E+09	0,006000000166893
8d	5,03E+09	0,00799999997019768
10d	7,85E+09	0,00999999977350235
12d	0,000113000009249449	0,012000000333786
14d	0,000154000010818839	0,0140000008940697
16d	0,000201000015431643	0,0159999999403954
20d	0,000314000005453825	0,0199999995470047
25d	0,000491000008711815	0,0250000001907349
26d	0,000531000029001236	0,0259999997138977
28d	0,000616000043275356	0,0280000017881393
N12	0,000113000009249449	0,012000000333786
N16	0,000201000015431643	0,0159999999403954
N20	0,000314000005453825	0,0199999995470047
N24	0,000452000036997795	0,024000000667572
N28	0,000616000043275356	0,0280000017881393
N32	0,00080400006172657	0,0319999998807907
N36	0,00102000001747131	0,036000001001358

ÖZGEÇMİŞ

Adı-Soyadı : Ekrem Tüfekçi
Doğum Tarihi ve Yeri: 29/02/1992 İSTANBUL
E-posta : et@ekremtufekci.com

Öğrenim Durumu :

- **Lisans** : 2017, İstanbul Aydın Üniversitesi Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü

