

YAPILARIN DOĞRUSAL OLMAYAN DEPREM ANALİZLERİNDE BÜNYE FONKSİYONLARININ PARALELLEŞTİRİLMESİ

B. Erkuş¹, B. Kasapoğlu² ve F. Yıldız²

¹Yrd. Doç. Dr., İnşaat Müh. Bölümü, İstanbul Teknik Üniversitesi, İstanbul

²Yük. Lis. Öğr., İnşaat Müh. Bölümü, İstanbul Teknik Üniversitesi, İstanbul

E-posta: bariserkus@itu.edu.tr

ÖZET:

Bu bildiride, yapıların zaman-tanım alanında yapılan doğrusal olmayan deprem analizlerinde doğrusal olmayan malzeme bünye-model fonksiyonlarının paralelleştirilmesi incelenmiştir. Son yıllarda, yapısal analizler için kullanılan ticari ve bilimsel yazılımlar, işlemci mimarisindeki gelişmeler sayesinde, kişisel bilgisayarlarda paralel işleme yöntemlerini uygulayabilir hale gelmiştir. Bu yöntemler genellikle, analizlerde en fazla süre gerektiren doğrusal denklem takımı çözümü gibi matris işlemlerini hızlandırmak için kullanılmaktadır. Doğrusal olmayan deprem analizlerinde, bu işlemlere ek olarak, özellikle fiber model gibi yayılı plastisite modellerinin kullanılması durumunda çok sayıda bünye-model fonksiyonlarının hesaplaması gerekebilmektedir. Bu fonksiyonların sayısının artması durumunda, hesaplarını hızlandırmak amacı ile paralel programlama yöntemleri incelenmiştir. Programlama dilleri olarak, son yıllarda bilimsel araştırmalarda kullanımı artan ve betik dil kabul edilen MATLAB ve Python kullanılmıştır. MATLAB programla dilinde analiz sürelerini etkileyen doğal ek süreler incelenmiş ve paralel programlanın etkin olması için gerekli olan minimum doğrusal olmayan eleman sayısı araştırılmıştır. Python dilinde, çubuk elemanlardan oluşan iki boyutlu bir çerçevenin doğrusal olmayan analizini yapabilen ardışık programlamaya dayalı birer program geliştirilmiştir. Doğrusal olmayan davranış, doğrusal çubuk elemanların uçlarında bulunan çift-doğrusal ve Takeda modellerini kullanan moment-dönme yayları ile modellenmiştir. Hareket denklemlerinin çözümü için Newmark- β ve dengelenmemiş kuvvet düzeltme yöntemleri kullanılmıştır. Her biri dört açıklıklı olan 10, 25, 50, 75 ve 100 katlı beş adet çerçeve örnek olarak oluşturulmuştur. Normalde ardışık olan bünye-model fonksiyon hesaplama döngüsü, kullanılan diller tarafından sağlanan araçlar ile paralelleştirilmiştir. Bünye-model fonksiyonlarının ardışık ve paralel hesaplanması durumunda sadece bu hesaplar için gerekli olan süreler bir deprem kaydı için elde edilerek karşılaştırılmıştır. Sonuçlar, fonksiyon sayısının artması ve hesap süresi daha fazla vakit alan Takeda modelinin kullanılması durumunda paralelleştirmenin daha etkili olduğunu göstermektedir. Ancak, incelenen örneklerde doğrusal olmayan eleman sayısının az olmasından dolayı, paralelleştirmenin tüm analiz süresine katkısı da az olmuştur.

ANAHTAR KELİMELER : Paralel programlama, doğrusal olmayan analiz, bünye-model fonksiyonları, deprem, MATLAB, Python

PARALELLIZATION OF CONSTITUTIVE FUNCTIONS IN THE NONLINEAR SEISMIC ANALYSIS OF STRUCTURES

ABSTRACT:

In this paper, parallelization of material nonlinear constitutive-model functions in the nonlinear seismic analysis of structures is investigated. In the recent years, commercial and academic structural analysis programs are able to implement parallel processing methods in the personal computers due to the recent developments in the architecture of processors. These methods are applied to speed-up the matrix operations that have the longest durations, such as solution of linear system of equations. In the nonlinear seismic analyses, simulation of large number of constitutive-model functions may be required in addition to the matrix operations, particularly for structures with distributed plasticity elements such as fiber elements. Parallel programming is investigated to speed-up the simulations in the case that large number of such elements are used. As programming languages, MATLAB and Python are used that are script languages and that are frequently used for research purposed in the recent years. In MATLAB, overhead that impact the analyses

durations is investigated and minimum number of nonlinear elements that is required for parallel programming to be effective is investigated. In Python, a program is developed that is based on sequential programming and that can analyze structures with frame elements. Nonlinear behavior are modelled with moment-rotation springs with bilinear and Takeda hysteresis that are placed at the end of frame elements. Newmark- β and unbalanced force correction method are used for the solution of equation of motion. Five frame structures with 10, 25, 50, 75, 100 stories each with four bays are established as examples. Constitutive-model function call loop that is originally programmed as serial is parallelized with the tools provided by the programming language. Durations for only the model function calls are obtained and compared for an historical ground motion data. Results show that parallel processing is more effective, when larger number of nonlinear elements are used or Takeda model that takes longer call times is used. On the other hand, contribution to the overall simulation is minor since number of nonlinear elements are limited in the sample structures investigated.

KEYWORDS: Parallel programming, nonlinear analysis, constitutive-model functions, seismic, MATLAB, Python

1. GİRİŞ

Doğrusal olmayan analizler (DOAler), deprem yükleri altındaki yapıların davranışını en iyi şekilde yansıtılabilen yöntemler olarak kabul edilmektedir. Performansa dayalı tasarımda DOAler, doğrusal analizlere göre yapı hasar ve performansını daha iyi yansıttıklarından dolayı mühendislik alanında tercih edilmektedir. Bilimsel araştırmalarda ise her malzeme ve yapının doğrusal olmayan davranışının deneyler ile incelenmesi mümkün olmadığından, DOAlere ihtiyaç vardır.

DOAlerde, en fazla kullanılan doğrusal olmayan modellerin başında yığılı plastik modeller gelmektedir. Yığılı plastik modellerde mukavemet ve rijitlik bozulma modları bir makro model ile ifade edilir. Yığılı plastik modellerin bir makro model olmasından dolayı analiz süresi hızlıdır ancak, doğrusal olmayan modlar açık olarak modellenmediğinden dolayı öngörülen davranış dışında kullanılmaları zordur. Yığılı plastik modeller eleman bazında analiz yapılacaksa iyi bir kalibrasyon ile doğruluğu yüksek sonuçlar verebilir. Ancak, tüm yapı analizlerinde kalibrasyonları çok zor veya imkansız olmasından ve modelin öngöremediği etkilerin oluşmasından dolayı doğruluk azalmaktadır. Bu durumda problemin doğasına bağlı olarak doğruluğu daha yüksek olan modelleme ve analiz yöntemlerinin kullanılması tercih edilmektedir.

Yayılı plastik modellerin, bozulma modları açık olarak modellendiğinden dolayı doğru kalibrasyon ile yığılı modellere göre doğruluğu daha yüksek sonuç verdikleri kabul edilir. Yayılı plastik modellere örnek olarak fiber modeller gösterilebilir. Fiber modellerde kesit birçok fibere bölünmekte ve bu fiberlerin her birisi ile malzemenin doğrusal olmayan davranışını temsil eden boyutsuz yaylar ifade edilmektedir. Bundan dolayı yığılı modelde bir adet olan doğrusal olmayan elemanı, yayılı modelde onlar hatta yüzler mertebesinde doğrusal olmayan eleman ile ifade etmek gerekir. Pratik mühendislikteki yaklaşım, büyük ölçekli yapılarda çubuk elemanlar için yığılı modeller kullanılması, yayılı modellerin ise, doğrusal olmayan davranışın en fazla beklendiği noktalarda kullanılması yönündedir. Örnek olarak, yüksek binalarda, kolon ve kirişlerde yığılı modeller kullanılırken, yayılı modeller sadece betonarme perde duvarlar için doğrusal olmayan davranışın beklendiği alt katlarda kullanılmaktadır. Bilimsel çalışmalarda ise hem eleman hem de tüm yapı analizleri için hertürlü elemanda davranışı daha iyi yansıttığı kabul edilen yayılı modeller kullanılabilir.

DOAlerdeki en önemli zorluk hata payı kabul edilebilir seviyedeki analizlerin zaman bakımından pahalı olmasıdır. Bunun bir nedeni, aynı zamanda doğrusal analizler ve benzer mühendislik problemleri için de geçerli olan yüksek ölçekli sistemlerde matris operasyonlarının fazla vakit almasıdır. Bu operasyonların başında doğrusal denklem takımı çözümü gelmektedir. Büyük ölçekli sistemlerde elde edilen denklem takımları ve bunları ifade eden matrisler çok büyük olmaktadır. Özellikle deprem analizi gibi her zaman anı için denklem takımı çözümünün gerekli olduğu problemlerde bu süre oldukça yüksek olmaktadır. Bu problemten dolayı birçok durumda kabul edilebilir analiz süreleri elde edebilmek için modelin basitleştirilmesi gerekmektedir. Bu da analiz sonuçlarının hata payını yükseltmektedir.

DOAlerin fazla vakit alıyor olmasının bir diğer nedeni, doğruluğu arttırmak maksadı ile fazla sayıda doğrusal olmayan eleman modelinin kullanılmasıdır. DOAlerde dengelenmemiş kesit tesirlerinin azaltılması amacıyla Newton-Rapson gibi yinelemeli yöntemler kullanılmaktadır. Bu yöntemlerdeki her yinelemede tekrardan

doğrusal denklem takımının çözümü söz konusudur. Eğer doğrusalsızlık yüksek ise, bu yinelemeler fazla olmakta ve bu durum analiz süresini oldukça arttırmaktadır. Doğrusal olmayan elemanların fazla olması durumunda analiz sürelerini azaltmak amacıyla doğrusalsızlığı parçalı doğrusal olarak ifade etmek ve/veya yinelemeli yöntemler yerine dengelenmemiş kuvvet düzeltmesi gibi yinelemesiz yöntemler kullanılabilir. Ancak yapılan bilimsel çalışmalarda bu yaklaşımlar ile analiz süreleri azalırken hata paylarında artış gözlemlenmiştir.

DOAlerde süreleri azaltmak maksadı ile kullanılan en önemli yaklaşım paralel programlamadır. Paralel programlama genellikle doğrusal denklem takımı çözümü için kullanılmaktadır. Doğrusal denklem takımı çözümü birçok mühendislik alanında karşılaşılan ortak bir problem olduğundan dolayı özellikle matematik ve bilgisayar alanında çalışan araştırmacılar bu ortak problemin çözümüne yönelik, doğrusal denklem takımı çözüm sürelerini paralel programlama ile kısaltmışlardır (Intel, 2017; NAG, 2017). Bu yöntemlerde ana amaç büyük denklem takımını daha küçük takımlar ile ifade edip işlemcileri eş zamanlı paralel bir biçimde kullanarak toplam denklem takımının çözüm süresini kısaltmaktır. Öyle ki, şu anda bilgisayar işlemci teknolojisi paralel ve yayılı işlemeye daha uygun olan işlemcilerin ya da sadece bu iş için üretilen özel bileşenlerin geliştirilmesi yönünden ilerlemektedir. Günümüzde tipik masaüstü bilgisayarlar bu işlemleri standart olarak yapacak şekilde üretilmektedir. Sonuç olarak, DOAlerde denklem takımının çözümü paralel programlama ile hızlandırılabilir.

İnşaat mühendisliğinde uygulamalar için bazı örnekler verilebilir. Frank ve Gregory (2000) tarafından yapılan çalışmada nesne tabanlı programlama ile doğrusal olmayan analiz, alan ayrıştırması (domain decomposition) yöntemi ile paralelleştirilmiştir. Bu çalışma OpenSees programının (McKenna, 1997) paralel işleme alt yapısını oluşturmuştur “OpenSeesSP” ve “OpenSeesMP” olmak üzere iki farklı paralel derleyici vardır. “OpenSeesSP” çok büyük modellerin analizlerinde, “OpenSeesMP” birden fazla küçük ölçekte birçok modelin paralel analizinde kullanılmaktadır (McKenna ve Fenves, 2008). Diğer bir çalışmada Dizon (2016) tarafından geliştirilen “FRAME3D” çelik yapı doğrusal olmayan analiz programının 2016 sürümünde hibrit paralel programlama çözümü uygulanmıştır. Bu yöntemlerde ana amaç büyük denklem takımını daha küçük takımlar ile ifade edip işlemcileri eş zamanlı kullanarak toplam denklem takımının çözüm süresini kısaltmaktır.

DOAlere özel olan ancak diğer mühendislik/matematik problemleri ile ortak olmayan bir konu doğrusal olmayan eleman sayısı ile ilgilidir. Bu durum özellikle son yıllarda geliştirilen karmaşık yayılı plastisite modelleri için daha belirgindir. Yukarıda anlatıldığı üzere, yığılı modellerde bir adet doğrusal olmayan eleman, yayılı modellerde çok daha fazla doğrusal olmayan elemana denk gelmektedir. Ayrıca, yayılı modellerde sadece yapı bazında değil eleman bazında da dengelenmemiş kuvvetlerin azaltılması gerekmektedir. Yayılı da olsa yığılı da olsa, özellikle yüksek doğruluk hedefi ile çok sayıda doğrusal olmayan elemanın kullanılması durumunda bu elemanların bünye denklemlerinin simülasyonu gerekmektedir. Bu simülasyonlar özellikle sürekli ve differensiyel denklem çözümü gerektiren modellerde vakit alıcı olabilmektedir. Sonuç olarak, bünye denklemlerinin fazla olması ya da yayılı modellerin kullanılması durumunda simülasyonlarının analiz süresine etkisi ya da hızlandırılması üzerine literatürde bir araştırma bulunmamaktadır.

Bünye denklemlerinin simülasyonlarının ve/veya yayılı modellerde dengelenmemiş kuvvetlerin azaltılması için yinelemeli analizlerin hızlandırılması amacıyla paralel programlama bir yöntem olarak kullanılabilir. Yayılı model kullanmak yerine aynı etkiyi gözlemlenmek amacıyla yığılı eleman sayısı fazla olan modeller kullanılarak simülasyonların sayısı artırılabilir. Bu durumda tüm sistem çözüm süresi fazla olacağından sadece bünye denklem simülasyon süresi incelenebilir. Mevcut analiz programlarında genellikle bu simülasyonlar seri olarak yapılmaktadır. Bu işlemlerin seri yapılması yerine eldeki işlemci çekirdeklerine paralel olarak yaptırılabilir. Bu noktada iki yöntem öne çıkmaktadır. Bu yöntemlerden biri veri paralelleştirmesi bir diğeri ise işlem paralelleştirmesidir. İşlem paralelleştirmesinde, büyük bir işlem küçük işlemcilerle (iplik) ayrılır ve paralel bir biçimde farklı işlemci çekirdeklerinde çalıştırılır. Veri paralelleştirmesinde ise büyük bir veri işlemci çekirdek sayısına göre parçalara ayrılarak her bir çekirdekte aynı işlem farklı veriler için çalıştırılır. Böylelikle büyük ölçekli bir problemin çözüm süreci, problemi parçalara ayırarak ve bu parçaların eş zamanlı işlenmesi sağlanarak hızlandırılmış olur.

Bu makalede, DOAde bünye denklemlerinin paralelleştirilmesi ve yapısal analiz sürelerine etkisi incelenmiştir. Bu amaçla, son yıllarda bilimsel araştırmalarda kullanımı artan ve betik dil kabul edilen Python ve MATLAB kullanılmıştır. Her iki dilde, çubuk elemanlardan oluşan iki boyutlu bir çerçevenin DOAini

yapabilen ardışık programlamaya dayalı birer program geliştirilmiştir. Doğrusal olmayan davranış, doğrusal çubuk elemanların uçlarında bulunan çift-doğrusal modeli kullanılarak moment-dönme yayları ile modellenmiştir. Hareket denklemlerinin çözümü için Newmark- β ve dengelenmemiş kuvvet düzeltme yöntemleri kullanılmıştır. Her biri 4 açıklıklı olan 10, 25, 50, 75 ve 100 katlı 5 adet çerçeve örnekleri incelenmiştir. Normalde ardışık olan bünye fonksiyon hesaplama döngüsü, kullanılan diller tarafından sağlanan araçlar ile paralelleştirilmiştir. Bünye fonksiyonlarının ardışık ve paralel hesaplanması durumunda sadece bu hesaplar için gerekli olan süreler bir deprem kaydı için elde edilerek karşılaştırılmıştır.

2. MATEMATİKSEL BİLGİLER

Bu bölümde, bu makalede kullanılan yapı modeli, modelin hareket denklemleri, hareket denklemlerinin çözüm yöntemi ve betik dillerdeki paralel programlama yöntemleri açıklanmıştır.

2.1. Yapısal Modelleme

Bu makalede kullanılan yapısal model, tipik çubuk elemanlardan ve dönme yaylarından oluşan bir çerçeve sistemdir (Şekil 1). Dönme yayları kolon ve giriş yüzlerine yerleştirilmiştir ve boyları sıfırdır. Tüm yapının doğrusal olması durumunda kuvvet, rijitlik ve yerdeğiştirme matrislerini içeren statik denge denklemleri yerdeğiştirme yöntemleri ile elde edilebilir. Dinamik denge denklemleri de benzer yöntemler ile hareketli serbestlik dereceleri için elde edilebilir. Doğrusal olmayan modelde, dönme yayaları kesitlerin moment-eğrilik ilişkilerinin çift-doğrusal hale getirilmesi ile elde edilmiştir.

2.2. Doğrusal Olmayan Analiz (DOA) Yöntemi

Bu makalede DOA için Newmark- β (Newmark, 1959) ve dengelenmiş kuvvet düzeltme (Powell, 1973) yöntemleri kullanılmıştır. Tüm yöntemin detayları Erkus (2004)'da verilmiş olup burada kısaca özetlenmiştir. Tüm yapının hareketli serbestlik dereceleri için dinamik denge denklemi şu şekilde yazılabilir:

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) + \mathbf{F}_s(t) = \mathbf{P}(t) \quad (1)$$

Burada, $\mathbf{x}(t)$, zemine göre rölatif olan yerdeğiştirmeleri ve dönmeleri içeren vektör, \mathbf{M} kütle matrisi, \mathbf{C} , sönümleme matrisi, \mathbf{K} doğrusal elemanlardan gelen rijitlik matrisi, $\mathbf{F}_s(t)$ doğrusal olmayan eleman kuvvetleri ve $\mathbf{P}(t)$ dış kuvvet vektörüdür. Bu denklemin t anındaki hali ve $t + \Delta t$ anındaki halinden çıkarılırsa aşağıdaki denklem elde edilir:

$$\mathbf{M}\Delta\ddot{\mathbf{x}}' + \mathbf{C}\Delta\dot{\mathbf{x}}' + \mathbf{K}\Delta\mathbf{x}' + \Delta\mathbf{F}_s' = \Delta\mathbf{P}' \quad (2)$$

Burada $\Delta[\]'$, büyüklüğün $t + \Delta t$ ve t anlarındaki değerleri arasındaki farklı göstermektedir. Hareket denklemi Newmark- β yöntemi ile aşağıda gösterilen artımsal ve cebirsel forma çevrilebilir:

$$\mathbf{A}\Delta\mathbf{x}' + \Delta\mathbf{F}_s' = \Delta\hat{\mathbf{P}}', \quad \mathbf{A} = \frac{\beta}{\beta\Delta t^2}\mathbf{M} + \frac{\gamma}{\beta\Delta t}\mathbf{C} + \mathbf{K}, \quad \Delta\hat{\mathbf{P}}' = \Delta\mathbf{P}' + \left(\frac{1}{\beta\Delta t}\mathbf{M} + \frac{\gamma}{\beta}\mathbf{C}\right)\dot{\mathbf{x}}' + \left[\frac{1}{2\beta}\mathbf{M} + \Delta t\left(\frac{\gamma}{2\beta} - 1\right)\mathbf{C}\right]\ddot{\mathbf{x}}' \quad (3)$$

Burada, β ve γ Newmark parametreleridir. Analizde, her zaman adımında artımsal doğrusal olmayan kuvvet için bir kabul yapılır. Bu kabul, doğrusal olmayan elemanların tanjant rijitliği üzerinden olabilir. Bu durumda

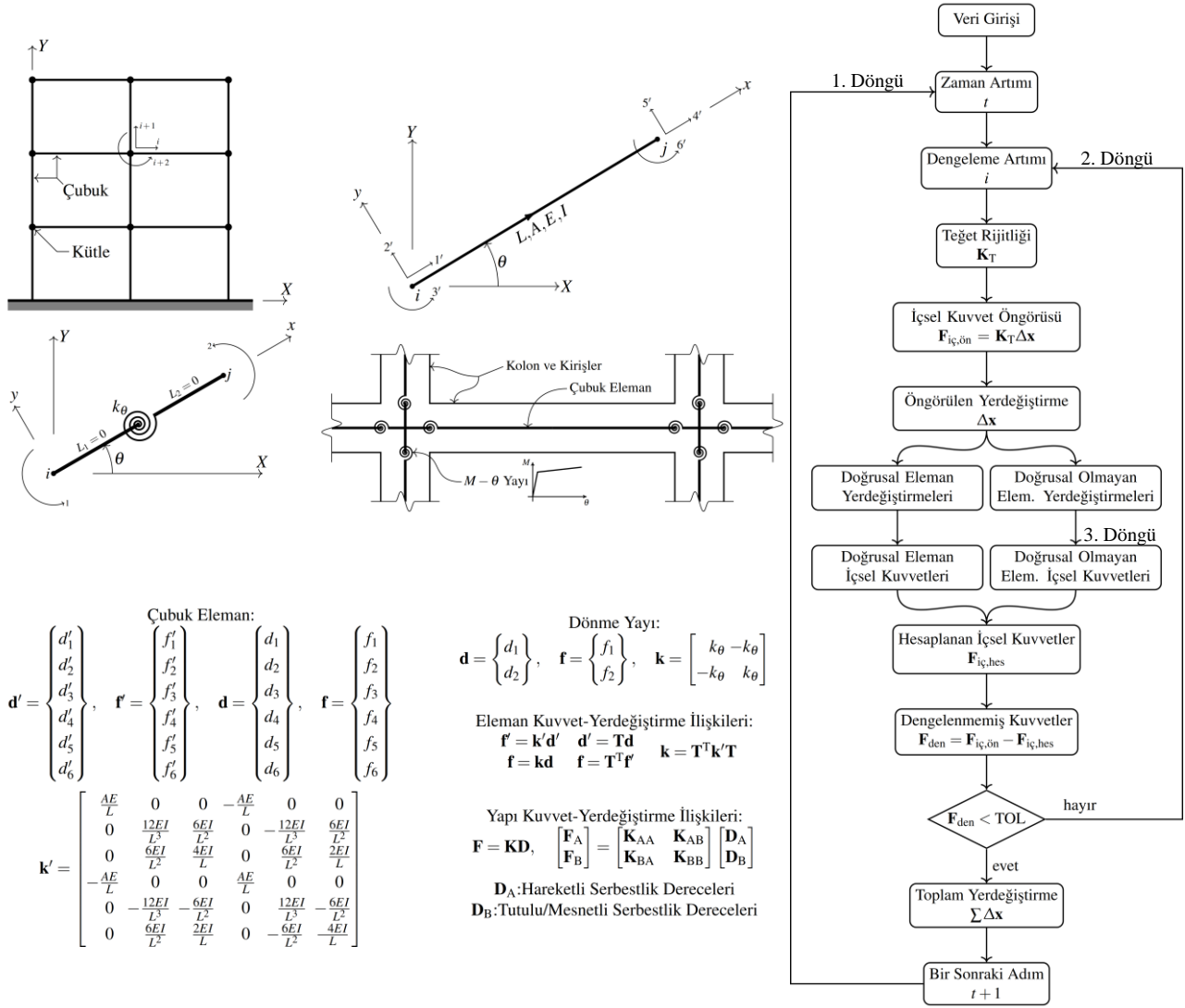
$$\Delta\mathbf{F}_s^{r,kabul} = \mathbf{K}_T\Delta\mathbf{x}^{r,kabul} \quad \text{ve} \quad (\mathbf{A} + \mathbf{K}_T)\Delta\mathbf{x}^{r,kabul} = \Delta\hat{\mathbf{P}}' \quad (4)$$

olur. Kabul edilen yer değiştirme için, bünye fonksiyonlarından doğrusal kuvvet $\Delta\mathbf{F}_s^{r,\text{çiftdoğ}}$ hesaplanabilir. Bu durumda, t adımı için dengelenmemiş kuvvet şu şekilde olur:

$$\Delta\mathbf{F}_s^{r,denge} = \Delta\mathbf{F}_s^{r,kabul} - \Delta\mathbf{F}_s^{r,\text{çiftdoğ}} \quad (5)$$

Dengelenmemiş kuvvet bir sonraki zaman adımında ek dış kuvvet olarak yapıya etkilerek sistem çözülür:

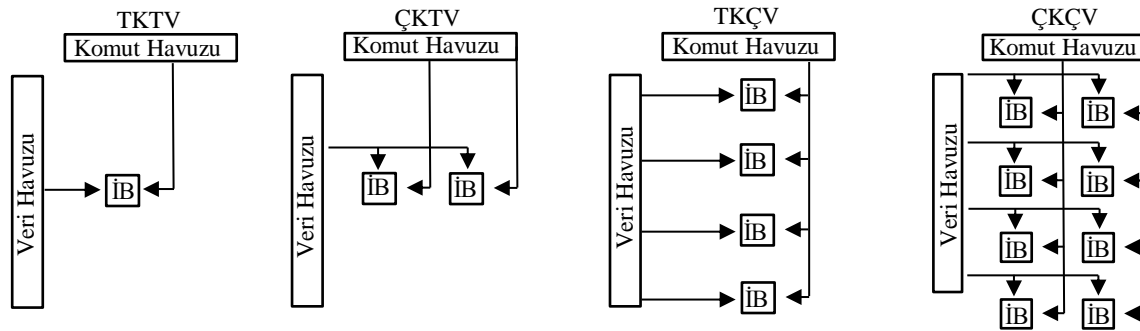
$$\mathbf{A}\Delta\mathbf{x}^{r+1} + \Delta\mathbf{F}_s^{r+1} = \Delta\hat{\mathbf{P}}^{r+1} + \Delta\mathbf{F}_s^{r,denge} \quad (6)$$



Şekil 1. Yapı, matematiksel model özetleri ve DOA akış şeması

2.3. Paralel Programlama

Konvansiyonel seri programlamada birden fazla işlem sıra ile tek işlemcide çalıştırılır, buna. Paralel programlamada, birden fazla işlem küçük parçalara ayrılarak birden fazla işlemcide veya bir işlemcinin birden fazla çekirdeğinde koşturulur (Şekil 2).



İB: İşlemci birimi veya çekirdek

Şekil 2. Seri ve paralel programlama yöntemlerinin şematik gösterimi

Paralel bilgisayarlar birçok şekilde sınıflandırılabilir. Bunlardan biri olan komut ve veri akışına dayalı sınıflandırma, veri ve komut arasındaki ilişkiyi göz önünde bulundurur. Flynn (1966) tarafından yapılan klasik sınıflandırma ile dört farklı grup oluşmaktadır.

1. **Tek komut tek veri (TKTV):** Bu bilgisayar yapısı tek işlemcilerdir ve tek bir veri akışı üzerinde işlem yapan tek bir komutu sıralı biçimde çalıştırır. Geleneksel tek işlemcili bilgisayarlar buna örnek verilebilir.
2. **Tek komut çoklu veri (TKÇV):** Bu bilgisayarlar birden çok özdeş işlemciden oluşur ve bu işlemcilerin kendi yerel hafızası bulunur. Her bir işlemciye bir tane gelecek şekilde birden fazla veri vardır. Eş zamanlı olarak, bütün işlemciler tek bir komut ile farklı veri grupları üzerinde çalışırlar. Bu, veri seviyesindeki paralelleştirmeye örnektir.
3. **Çoklu komut tek veri (ÇKTV):** Bu modeller birden çok işlemciden oluşur. Her işlemcinin kendine ait yerel bir kontrol birimi vardır ve işlemciler tek bir hafıza birimini paylaşırlar. Bütün işlemciler hafızadan aldıkları aynı veriyi eş zamanlı olarak kontrol birimlerinden aldıkları farklı komutlar ile işlerler. Bu, komut seviyesindeki paralelleştirmeye örnektir.
4. **Çoklu komut çoklu veri (ÇKÇV):** Bu bilgisayar yapısı en genel ve diğerlerine göre daha güçlü bir yapıdır. Birden çok işlemci, birden çok komutu birden çok veri üzerinde çalıştırabilir. Her işlemcinin kendine ait kontrol ve hafıza biriminin bulunması bu bilgisayar modelini kuvvetli hale getirir. Bu bilgisayar mimarisi kişisel bilgisayarlara, süper bilgisayarlara ve bilgisayar çalışma ağlarına uygulanabilir.

Python’da Paralel Programlama

Python’da paralel programlamaya uygun iki önemli kütüphane vardır. Bunlardan biri “threading”, bir diğeri “multiprocessing” kütüphanesidir. “threading” kütüphanesi ile girdi ve çıktı operasyonlarını (I/O) paralelleştirmek uygun olurken, matematiksel işlemlerin kullanıldığı fonksiyonlarda “GIL” klidi sebebiyle uygun olmamaktadır. Bu sebeple bu makalede paralel programlama “multiprocessing” kütüphanesinin “Pool” sınıfı ve “map()” fonksiyonu kullanılarak gerçekleştirilmiştir (Şekil 3). “n” sayıda elemanın “n” sayıda yerdeğiştirmesi 6 (çekirdek sayısı) gruba ayrılmıştır ve her bir işlemcide bir grup elemanın yerdeğiştirmelerinin oluşturduğu içsel kuvvetler çift-doğrusal model fonksiyonu yardımıyla hesaplanmıştır.

```
if __name__ == "__main__":  
    p = Pool(processes=6)  
    içsel_kuvvetler =  
        p.map(çift_doğrusal_fonksiyon,  
            [yerdeğiştirme[0:n/6],  
             yerdeğiştirme[n/6: 2n/6],  
             yerdeğiştirme[2n/6:3n/6],  
             yerdeğiştirme[3n/6:4n/6],  
             yerdeğiştirme[4n/6:5n/6],  
             yerdeğiştirme [5n/6:n] ])
```

Şekil 3. “multiprocessing”
kütüphanesinin kullanımı

MATLAB’da Paralel Programlama

Bu çalışmada MATLAB programlama dilinin “parfor” ve “spmd” fonksiyonları kullanılmıştır. “parfor”, çalışması süresi fazla bir fonksiyonun bir döngü içinde çok defa çağırılması durumunda etkili olmakta (örnek: Monte Carlo simülasyonu) ve zaman bakımından kazanım sağlamaktadır. Ancak, iç içe döngülerin olduğu durumlarda doğal ek süre (overhead) nedeni ile tüm süre açısından “parfor” fonksiyonunun etkinliği azalmaktadır. İç içe olan döngülerde, “spmd” fonksiyonu daha faydalı olabilmektedir. Tek işlem çok veri olduğu durumlarda bu fonksiyon kullanılarak veri parçalara ayrılabilir ve aynı işlem farklı işlemcilerde farklı veriler için eş zamanlı çalıştırılarak paralel programlama yapılabilir. Ancak algoritma uygun değilse, burada da en büyük zorluk doğal ek süre etkisi olabilmektedir.

Doğal Ek Süre (Overhead) Etkisi

Paralel programlamada karşılaşılan bir konu verinin işlenmesi için gerekli olan süreden farklı olarak paralel programlamanın doğasında olan işlemlerden dolayı oluşan ek süredir. Örnek olarak MATLAB paralel programlama için istemci (client) hafıza bölgesinden işçi (worker) tarafında veri aktarmaktadır. Bu aktarım için gerekli olan süre doğal ek süredir (overhead). Bu aktarım iç içe iki döngü nedeni ile birden fazla yapılması gerekiyorsa, doğal ek süre döngü sayısı kadar artacaktır ve paralel işlem ile kazanılan süreyi perdeleyebilir. Matematiksel olarak paralel ve seri işlem süreleri ve doğal ek süre ile ilgili koşul şu şekilde gösterilebilir:

$$t_p = t_{des} + t_{iş}(n_{iş}/n_ç), \quad t_s = n_{iş}t_{iş}, \quad t_p < t_s \quad (7)$$

Burada, t_p paralel işlem süresi, t_s seri işlem süresi, t_{des} paralel işleme özel doğal ek süre, $n_{i\dot{s}}$ işlem adeti ve n_c çekirdek ya da işlemci sayısıdır.

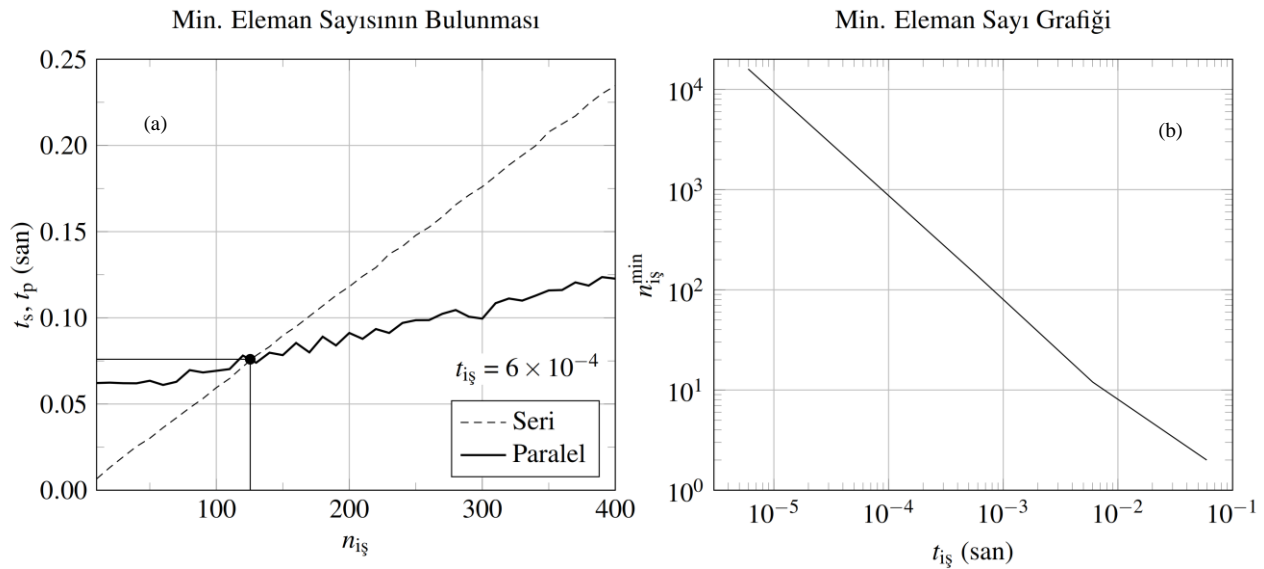
3. NÜMERİK ANALİZLER VE DEĞERLENDİRME

Bu bölümde, üç çalışma yapılmıştır. Birinci çalışmada MATLAB programında `parfor` ve `spmd` yaklaşımları için doğal ek süre çalışması yapılmıştır. İkinci çalışmada, örnek yapı Python programı ile seri ve paralel olarak çözülerek sonuçlar karşılaştırılmıştır.

3.1. MATLAB için Doğal Ek Süre Çalışması

Yapıların DOAlerinin programlanmasında genel de iç-içe olan iki ana döngü bulunmaktadır (bakınız Şekil 1). Bunlardan ilki zaman artım döngüsü (1. Döngü) diğer dengelenmemiş kuvvet döngüsünü azaltan döngüdür (2. Döngü). İkinci döngü içinde bünye fonksiyonlarının her eleman için çağrıldığı üçüncü bir döngü bulunmaktadır (3. Döngü). Bu makalede, ikinci döngü, kullanılan dengelenmemiş kuvvet düzeltme yöntemi nedeni ile mevcut değildir ve sadece 1. ve 3. Döngüler mevcuttur. MATLAB programında 3. Döngü paralelleştirildiği zaman 1. Döngünün her adımında 3. Döngü için doğal ek süre yaratmaktadır. Bundan dolayı, paralelleştirme işleminin seri işlemde daha hızlı olabilmesi için kullanılması gereken en az doğrusal olmayan eleman sayısının bilinmesi gerekir. Bu sayıyı bulmak için `parfor` ve `spmd` için iki ayrı çalışma yapılmıştır. Bu çalışmalar çekirdek sayısı 4 olan bir bilgisayarda gerçekleştirilmiştir.

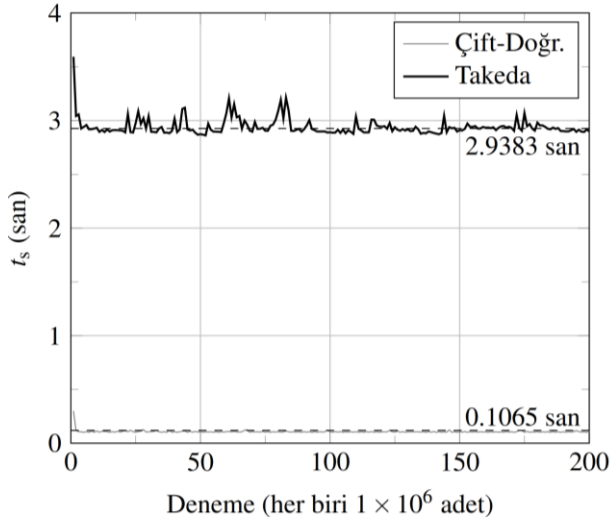
`parfor` için yapılan çalışmada farklı işlem süresi verebilen bir fonksiyon hazırlanmış, her bir işlem süresi için gerekli olan minimum çağrı sayısı belirlenmiştir. Bunun için ilk önce, verilen bir fonksiyon için seri çalışma hızını yakalayan fonksiyon çağrı sayısı belirlenmiştir (Şekil 4a). Bu sayı `parfor` da doğal ek süreyi yenmek için gerekli olana eleman sayısına denk gelmektedir. Bu çalışma farklı çağrı sürelerine sahip birçok fonksiyon için yapılmış ve minimum eleman sayı grafiği elde edilmiştir (Şekil 4a).



Şekil 4. MATLAB `parfor` yaklaşımı için bünye fonksiyonlarının çalışma sürelerinin ve doğal ek süreyi aşmak için gerekli minimum eleman sayısının belirlenmesi

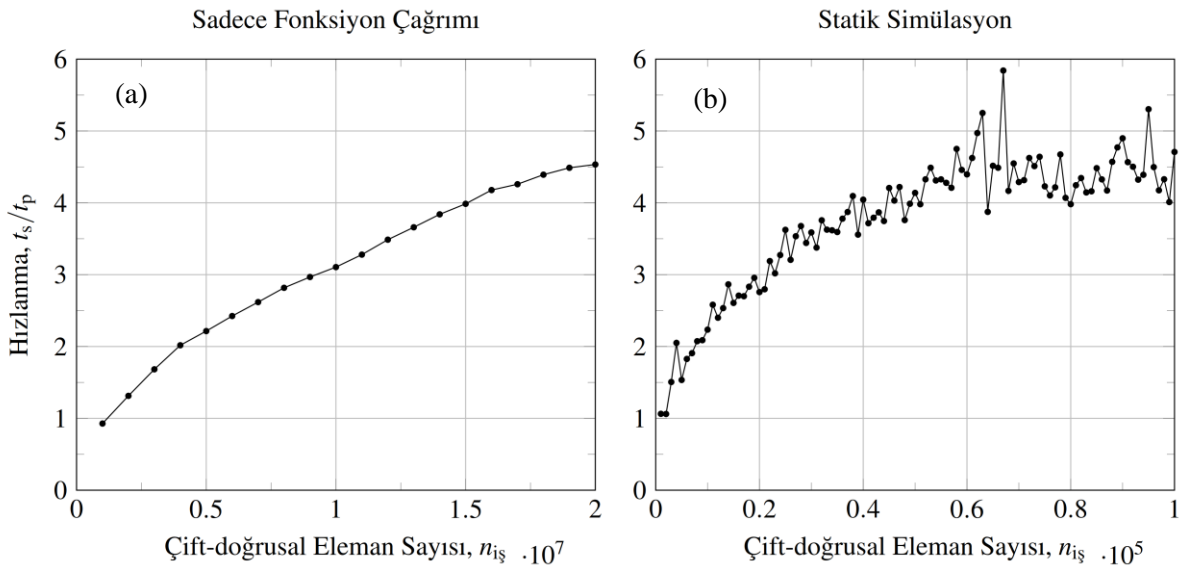
Örnek olması açısından çift-doğrusal ve Takeda fonksiyonları için işlem süreleri olarak hesaplanmıştır (Şekil 5). Bu hız değerlerine göre `parfor` ile paralel programların seri programlama ile aynı hızda ya da daha hızlı olması için yapıda yaklaşık olarak en az 10^6 adet çift doğrusal eleman veya en az 3×10^4 adet Takeda elemanı olması gerekmektedir. Bu değerler, minimum eleman sayı grafiğinin dış değerlemesi ile bulunmuştur. Bunun nedeni, `parfor` fonksiyonu nedeni ile oluşan doğal ek sürenin yüksek olmasıdır ki bu, `parfor` fonksiyonunun çift doğrusal ve Takeda elemanlarına çok uygun olmadığı anlamına gelmektedir. Bunun bir nedeni, MATLAB programının `if-else` yapısına sahip parçalı-doğrusal fonksiyonları çok hızlı çağırması olabilir.

İkinci olarak $spmd$ için yapılan çalışmada, MATLAB programının çift-doğrusal ve Takeda elemanları çok hızlı çalıştırmasından dolayı $spmd$ çerçeve yapısını analiz eden programında çağrılmamıştır; bu fonksiyonların çerçeve yapısında çağrılabilmesi için yapının çok büyük ölçekli olması gerekmektedir.



Şekil 5. Bünye fonksiyonları çalışma süreleri

Örnek olması için çift-doğrusal fonksiyonu $spmd$ döngüsü içerisinde programlanmış ve seri çağrıma göre hızlanması incelenmiştir (Şekil 6). Burada bünye fonksiyonlarının paralel ve seri çağrımları süreleri hesaplanmış ve seri çağrımlar için gerekli süre paralel çağrımlar süresine bölünerek hızlanma değerleri elde edilmiştir. Sadece fonksiyon çağrımında $spmd$ ile hızlanma yaklaşık olarak 10^6 adet çift doğrusal ile başlamıştır ve eleman sayısı arttıkça hızlanmada artmıştır (Şekil 6a). Bu sonuçlar *parfor* yönteminin aksine, $spmd$ ile paralel işlemin etkin bir şekilde gerçekleştiğini göstermektedir. Benzer şekilde çift-doğrusal elemanlar ile statik itme bir simülasyonu gerçekleştirilmesi durumunda hızlanma elde edilmiştir (Şekil 6b). Statik simülasyonlarda gerekli olan minimum eleman sayısının daha az olduğu gözlemlenmiştir.



Şekil 6. Çift doğrusal eleman için hızlanma değerleri

3.2. Python ile Doğrusal Olmayan Analizler

Python programı ile yapılan çalışmalarda, her biri 4 açıklıklı, açıklık genişliği 6 metre, kat yüksekliği 3 metre olan 10, 25, 50, 75 ve 100 katlı 5 adet çerçeve yapı modeli incelenmiştir (Tablo 1). Yapı kütlesi, bina ölü ağırlığına 30kN/m hareketli yük etkisi eklenerek hesaplanmıştır. Kolon kesitleri aksenal yüklerin kesit kapasitesinin yaklaşık olarak %35'ine denk gelecek şekilde belirlenmiştir. Kiriş kesitleri hareketli ve ölü yük etkisi altında güvenli tarafta kalacak şekilde belirlenmiştir. Donatı oranları yaklaşık olarak kolonda %1.2, kirişlerde %0.7'dir. Beton sınıfı C40, çelik sınıfı S420a'dır. XTRACT (2016) programı kullanılarak, her kesit için moment-eğrilik ilişkisi elde edilmiştir. Bu ilişkilerden yaklaşık olarak çift doğrusal idealleştirilmiş davranışlar belirlenmiştir. Dönme yayı elde edilirken doğrusal olmayan bölgenin boyunun eğilmeye çalışan kesitin yarısı olarak alınmıştır. Bu uzunluk kullanılarak moment-eğrilik ilişkisinden moment-dönme ilişkisine geçilmiştir. Bu modellerin analizleri için MATLAB ve Python'da programlar geliştirilmiştir.

Geliştirilen programların doğruluğu hem doğrusal hem de doğrusal olmayana analizlerde 10 katlı örnek bir yapı için doğruluğu genel kabul görmüş SAP2000 (2016), programı ile kontrol edilmiştir. SAP2000 modelin rijit-plastik mafsallar kullanılmış, benzer modeller olmaları için yazılan programda dönme yaylarının ilk rijitlikleri çok yüksek alınmıştır. Burada sadece DOA kontrolünün sonuçları verilmiştir (Şekil 7.).

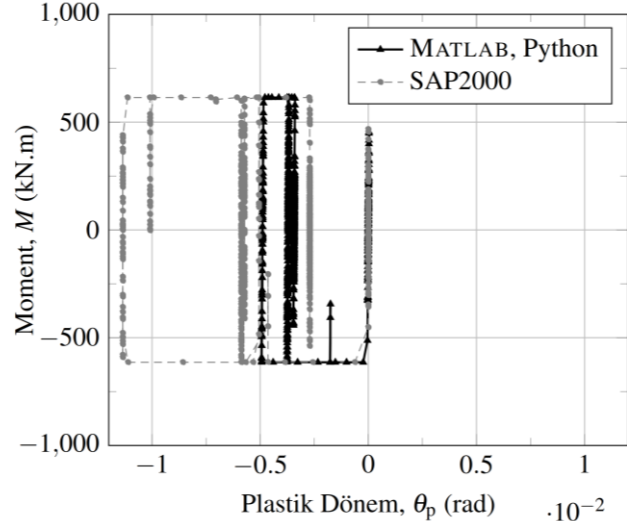
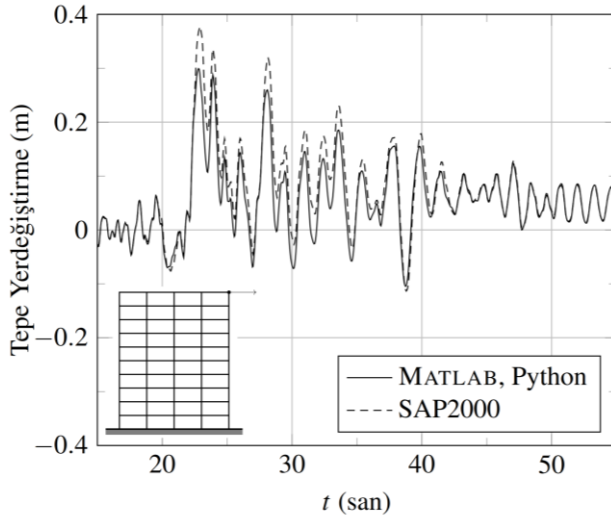
Görüldüğü üzere, yazılan program ile SAP2000 sonuçları benzer sonuçlar vermektedir. İki sonuç arasındaki farkın, kullanılan doğrusal olmayan eleman modellerindeki farklılıktan kaynaklandığı düşünülmekte olup, eleman parametrelerinin kalibrasyonu ile sonuçlar daha benzer hale getirilebilir. Ancak, bu makalede amaç bünye fonksiyonlarının paralelleştirilmesi olduğundan, ileri kalibrasyona gerek duyulmamıştır.

Örnek yapılar bünye fonksiyonlarının hem seri çağrıldığı hem de paralel çağrıldığı iki ayrı Python programı ile analize tabi tutulmuştur. Paralel programlama için “multiprocessing” kütüphanesi kullanılmıştır. Paralel programlama ile bünye fonksiyon çağrı toplam süresi seri programlama ile gereken süreye bölünerek paralel programlama ile elde edilen hızlanma elde edilmiştir (Şekil 8). Bu çalışmalar 6 çekirdekli bir bilgisayarda yapılmıştır.

Tablo 1. İncelenen yapılar hakkında bilgiler

Kat Adedi	Kat Aralığı	Kolon (cm)	Kolon $M-\theta$ Yayı		
			k_1 (kN/m)	k_2 (kN/m)	F_y (kN)
10	0-10	50 × 50	564764	6992	614
25	0-10	80 × 80	2408280	53625	2609
	11-20	60 × 60	979850	22990	1114
	21-25	50 × 50	564764	6992	614
50	0-10	120 × 120	8158218	288628	9155
	11-20	100 × 100	5022212	126240	5268
	21-30	85 × 85	2943367	70301	3225
	31-40	70 × 70	1500820	34745	1741
	41-50	50 × 50	564764	6992	614
75	0-10	150 × 150	15106199	536410	17425
	11-20	135 × 135	10571945	359188	12079
	21-30	120 × 120	8158218	288628	9155
	31-40	105 × 105	5457429	135436	5919
	41-50	90 × 90	3553944	81722	3752
	51-60	75 × 75	2006157	43600	2167
	61-70	60 × 60	979850	22990	1114
100	71-75	50 × 50	564764	6992	614
	0-10	190 × 190	30151722	1079757	35513
	11-20	175 × 175	22396216	781683	26044
	21-30	160 × 160	16932471	586438	20035
	31-40	145 × 145	13140236	4570921	15538
	41-50	130 × 130	9132451	305766	10812
	51-60	115 × 115	6861433	177859	7804
	61-70	100 × 100	5022212	126240	5268
	71-80	85 × 85	2943367	70301	3225
	81-90	70 × 70	1500820	34745	1741
91-100	50 × 50	564764	6992	614	

Tüm modellerde, kirişler 50 cm × 40 cm’dir ve $M-\theta$ yaylarında $k_1 = 101832$ kN/m, $k_2 = 1664$ kN/m, $F_y = 204$ kN’dur

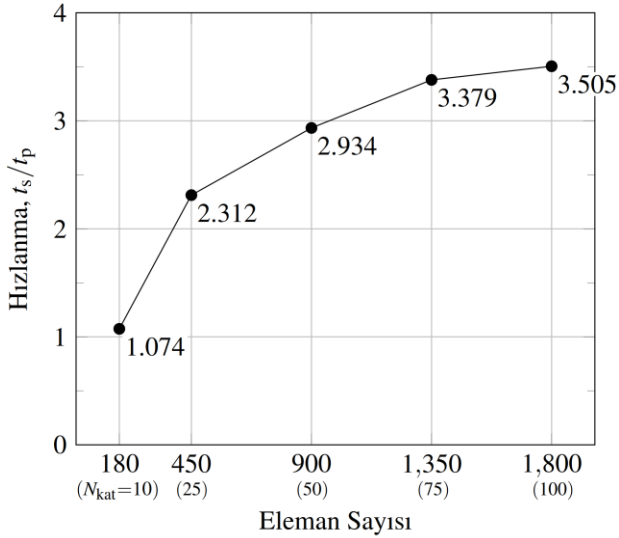


Şekil 7. MATLAB ve Python’da yazılan doğrusal olmayan analiz programının SAP2000 ile karşılaştırılması

Python’da elde edilen hızlanma değerleri, MATLAB’da elde edilen hızlanma değerlerine çok daha küçük sayıda çift-doğrusal eleman ile ulaşılmıştır. Bu sonuçlar, Python’da çift-doğrusal elemanların MATLAB’da olduğundan çok daha yavaş çalıştığını göstermektedir.

Python’da hem seri hem de paralel işleme ile yapılan analizler için tüm analiz süreleri ve çift-doğrusal eleman için çağrı süreleri ve paralel sürelerin seri sürelerle oranları Tablo 2’de gösterilmiştir. Bu tabloda ilk gözlemlenen tüm analiz sürelerinin bünye fonksiyon için gerekli sürelerle göre daha fazla vakit almasıdır.

Bunun birkaç nedeni bulunmaktadır. Doğrusal olmayan analizlerde doğrusal denklem takımlarının çözümü çok fazla süre almaktadır ve yazılan programlarda, programın bu bölümleri paralelleştirilmemiştir. Diğer bir neden ise doğrusal olmayan elemanların azlığıdır. Seçilen yapı örneğinde doğrusal eleman sayısı arttıkça, yapı serbestlik derecesi daha fazla artış göstermektedir. Bu ise denklem takımlarının çözümü için gerekli olan süreyi daha da arttırmaktadır. Bundan dolayı, bünye fonksiyonlarının paralelleştirilmesi doğrusal olmayan yay elemanların çok sayıda kullanıldığı ancak yapı serbestlik derecesinin fazla olmadığı fiber elemanlarının kullanımına daha uygun olduğu görülmektedir. Diğer bir konu, serbestlik derecesi ve doğrusal olmayan eleman sayısının arttıkça, tüm analiz sürelerinde artış olduğu gözlemlenmiştir. Bunun nedeninin doğal ek süre etkisi olduğu tahmin edilmekte olup çalışmanın ilerleyen aşamalarında daha detaylı araştırılması gerekmektedir.



Şekil 8. Python'da hızlanma

Tablo 2. Python'da toplam analiz ve çift-doğrusal eleman çağrım süreleri

Kat	Toplam Süre (san)			Bünye Süre (san)		
	Seri	Paralel	Seri/Par.	Seri	Paralel	Seri/Par.
10	1485	1402	94%	15.51	14.44	93%
25	6484	5962	92%	40.70	17.60	43%
50	20216	20052	99%	72.77	24.80	34%
75	45698	47180	103%	107.04	31.67	30%
100	85975	88994	104%	139.34	39.75	29%

4. SONUÇLAR

Bu makalede, yapıların zaman-tanım alanında doğrusal olmayan deprem analizlerinde bünye fonksiyonlarının paralelleştirilmesi araştırılmıştır. Bu amaçla Python ve MATLAB betik dilleri programlama dilleri olarak seçilmiştir. Dünya fonksiyonları olarak çift-doğrusal eleman ve Takeda modeli için hazırlanmış fonksiyonlar kullanılmıştır.

MATLAB dilinde bünye fonksiyonlarının çok hızlı çalışmalarından dolayı yapı analiz programında bünye fonksiyonlarının paralelleştirilmesi programlanmamıştır. MATLAB dilinde iki paralelleştirme yöntemi incelenmiştir. `parfor` yönteminde özellik doğal ek süre (overhead)'den dolayı doğrusal olmayan analizlerin algoritmasında bulunan iç içe döngü akışına uygun olmadığı belirlenmiştir. Doğal ek süreyi yenmek için gerekli olana minimum bünye fonksiyon sayılarını çift-doğrusal ve Takeda modelleri için belirlenmiştir. Diğer yöntem olan `spmd` yönteminde paralelleştirme gerçekleştirilmiştir ve sadece fonksiyon çağırımı için ve statik itme tipi analiz için hızlanma miktarları eleman sayısına bağlı olarak belirlenmiştir.

Python dilinde örnek bir çerçeve yapı tipi için zaman-tanım alanında doğrusal olmayan analiz yapabilen bir program geliştirilmiştir. Bu programda bünye fonksiyonları paralelleştirilerek bu fonksiyonlardaki hızlanma incelenmiştir. Python'da MATLAB diline göre daha az sayıda çift-doğrusal eleman ile hızlanma gerçekleşmiş, ancak bu hızlanma tüm süreye önemli bir katkı olarak yansımamıştır. Bunun nedeninin analizin diğer bölümlerinin paralelleştirilmemiş olması gösterilmektedir. Diğer bir gözlem ise doğal ek süre etkinin olduğu yöndedir.

KAYNAKLAR

Dizon, A. B. R. (2016). "A Hybrid-Parallel Framework for the Nonlinear Seismic Analysis of Very Tall Buildings." California Institute of Technology.

- Erkus, B. (2004). "Comparison of the techniques used in the Newmark analysis of nonlinear structures." Proc., 17th ASCE Engineering Mechanics Conference, Newark, Delaware.
- Flynn, M. J. (1966). "Very high-speed computing systems." Proceedings of the IEEE, 54(12), 1901-1909.
- Frank, M., ve Gregory, L. F. (2000). An Object-Oriented Software Design for Parallel Structural Analysis.
- Intel (2017). "Math Kernel Library (MKL)." <<https://www.intel.com>>. (July 2017, 2017).
- McKenna, F., ve Fenves, G. L. (2008). "Using the OpenSees interpreter on parallel computers." Network for earthquake engineering simulations.
- Mckenna, F. T. (1997). "Object-oriented finite element programming: frameworks for analysis, algorithms and parallel computing." Ph.D. Thesis, University of California, Berkeley.
- NAG (2017). "The NAG MPI Parallel Library." <<https://www.nag.com/>>. (July 2017, 2017).
- Newmark, N. M. (1959). "A Method of Computation for Structural Dynamics." Journal of the Engineering Mechanics Division, ASCE, 85(3), 67-94.
- Powell, G. H. (1973). "DRAIN-2D User's Guide." Report No: EERC 73-22. Earthquake Engineering Research Center, University of California, Berkeley, California.
- SAP2000. 2016. SAP2000, version v18, Computer and Structures, Inc., Berkeley, California.
- XTRACT. 2016. XTRACT, version v3, TRC, Rancho Cordova, California.