

## Çok-cisim integrasyonunda zaman-simetrik, ayrık blok zaman adımlı algoritma

**Murat KAPLAN\***, Hasan SAYGIN

İTÜ Bilişim Enstitüsü, Hesaplamalı Bilim ve Mühendislik Programı, 34469, Ayazağa, İstanbul

### Özet

*Ayrık blok zaman adımlı integrasyon algoritmaları, çok-cisim problemlerinde sıklıkla kullanılan algoritmalarındandır. Farklı zaman ölçeklerine sahip cisimlerin aynı problemin parçaları olduğu durumlarda, kullanımları kaçınılmazdır. Ancak kullanılan integrasyon şeması zaman-simetrik bile olsa, blok zaman adımları kullanıldığında zaman simetrisi bozulmaktadır. Zaman simetrisinin bozulmasının doğal sonucu, toplam enerji üzerindeki hata birikiminin doğrusal olarak artmasıdır. Bu çalışmada, zaman-simetrik blok zaman adımlı bir algoritma geliştirilmiştir. Geliştirilen algoritma, sabit zaman adımları kullanıldığında zaman-simetrik yapısını koruyan integrasyon şemalarının, değişken ayrık blok adımlı bir algoritma içerisinde de zaman simetrik yapısını korumasını sağlamaktadır. Hatta integrasyon şeması tam olarak zaman simetrik olmasa bile, geliştirilen algoritma ile yüksek enerji korunumu sağlandığı gözlenmiştir. Özellikle uzun süreli integrasyonlarda algoritmanın yüksek başarımlı belirlenmiştir. Zaman-simetrik ayrık blok zaman adımlı algoritma, bir yineleme yapısı içermektedir. Bu yinelemenin ayrık zaman adımlı integrasyon şemasıyla uygulanabilmesi için, çağ aralıkları (era) belirlenmiş, yineleme işlemleri bu çağ aralıkları üzerinden sağlanmıştır. Yapılan testlerde bu çağ aralıklarının büyüklüğünün hem yineleme sayısı, hem de bellek tüketimi açısından önemli olduğu gözlenmiştir. Çalışmanın bu safhasında, algoritma çağ aralıklarının değişken tutulabildiği bir yapıya kavuşturulmuştur. Algoritma Leapfrog integrasyon şeması kullanılarak geliştirilmiş, etkinliği daha yüksek dereceden integrasyon şemalarıyla da sınanmıştır. İçerdiği yineleme yapısı yüzünden ayrık zaman adımlı önceki algoritmalarından daha yüksek zaman ve bellek tüketimine sahiptir. Ancak toplam enerji üzerindeki hata birikimlerini büyük ölçüde azaltmaktadır.*

**Anahtar Kelimeler:** Çok-cisim problemi, sayısal integrasyon, ayrık blok-zaman adımları.

\*Yazışmaların yapılacağı yazar: Murat KAPLAN. mkaplan@be.itu.edu.tr; Tel: (212) 285 70 77.

Bu makale, birinci yazar tarafından İTÜ Bilişim Enstitüsü, Hesaplamalı Bilim ve Mühendislik Programı'nda tamamlanmış olan "Çok cisim integrasyonu için zaman simetrik ayrık blok zaman adımlı algoritma" adlı doktora tezinden hazırlanmıştır. Makale metni 05.11.2008 tarihinde dergiye ulaştırılmış, 07.01.2009 tarihinde basım kararı alınmıştır. Makale ile ilgili tartışmalar 31.01.2011 tarihine kadar dergiye gönderilmelidir.

## A time-symmetric individual block time step algorithm in n-body integration

### Extended abstract

The gravitational  $N$ -body problem can not be solved analytically except for two-body cases. While there are various approximate methods, such as Monte Carlo or Fokker-Planck method, the most accurate way to solve the  $N$ -body problem is by direct integration of the orbits of the  $N$  bodies. Direct integration of the  $N$ -body problem is a necessity for many cases such as star cluster simulations.

During the last fifteen years, two approaches have been put forward to improve numerical conservation of energy and other theoretically conserved quantities: symplectic integration schemes, where the simulated system is guaranteed to follow a slightly perturbed Hamiltonian system, and time-symmetric integration schemes, where the simulated system follows the same trajectory in phase space, when run backward or forward. Time-symmetric integration schemes share with symplectic schemes the property that their energy errors show a much better behavior than the case for generic integration schemes. Allowing adaptive time steps typically leads to a loss of symplecticity. In contrast, time symmetry can be easily maintained, at least for a continuous choice of time step size. An iterative scheme must be combined with individual block time step scheme to apply the new algorithm to the  $n$ -body problem effectively.

However, straightforward implementation of time-symmetry, translated to block time steps, faces significant hurdles for  $N$ -body problem. For example, iteration can lead to oscillatory behavior, and even when such behavior is suppressed, energy errors show a linear drift in time.

In this work it is aimed to construct a time symmetric block time step integration scheme. We first consider to analyze block time step scheme with time symmetrization procedure for the gravitational two-body problem. It is hard to generate an efficient algorithm for  $N$ -body problem without deep and clear understanding of this fundamental problem. However it is easy to correct the algorithm in Kepler problem. Besides, it is known that contributions to the energy error are largely generated by close encounters between two particles. For these reasons, two-body problem is preferred to develop the algorithm in the first instance.

The algorithm is developed and tested in two-body problem for accuracy and energy conservations in many different Kepler problem with different iteration numbers. In the process of the development work, leapfrog integration scheme which is the second order time-symmetry scheme is preferred. Also fourth, sixth, and eighth order Hermite integration schemes are used for test runs.

The major conceptual difficulty in designing a time-symmetric block time step scheme is the global context information that is needed, with extensions toward the future as well as the past. We present an approach that circumvents these problems. We divide and conquer: we split the total history of our simulation into a number of smaller periods, which we call every one as era. Each era extends a period in time equal to the largest allowed time step  $t_{max}$ , or to an integer multiple of  $t_{max}$ , whatever turns out to be the most convenient.

For such reasons, we have analyzed the era concept in more detail for time symmetrized block time steps. Our test results show that size of era must be chosen carefully for stable and robust integrations. This is important especially for long term simulations with highly desirable energy conservations. Era size is also important to avoid extra data storage and uselessly high number of iterations which require too much running times.

As a second gain of the work, we re-designed the previous scheme, and suggested dynamically changing size for era. With this scheme, iteration process of the integration can follow needs of the simulation adaptively. And, era size will be well-adjusted with physics of the problem with proper criterion.

Our basic integrator in the work is leapfrog scheme. It is a well known, second order, time symmetric integration scheme for fixed time steps. We also used fourth, sixth, and eighth order Hermite schemes to see the ability of the algorithm with higher order integration methods. We used different Kepler problems in two-body tests, and used Plummer model initial conditions for  $N$ -body tests. The algorithm does not show linear growing error with time symmetric integrations. It also shows high energy conservations with other integration schemes especially for long integration times.

**Keywords:**  $N$ -Body integration, time symmetry, individual block - time step, Hermite integration.

## **Giriş**

Kütle çekimsel çok-cisim probleminin ikiden fazla cisim için analitik bir çözümü yoktur. Özellikle uzun süreli çok-cisim benzetimlerinde (simülasyon) temel değişmezleri korumak önemlidir. Toplam enerji, yapılan benzetimlerin doğruluğu üzerinde en çok başvurulan büyüktür (Aarseth, 2003).

Öte yandan, doğrudan integrasyon şemaları algoritmanın tamamını kapsasın ya da kapsamasın, benzetim şemalarının en büyük zaman tüketimine sahip kısımlarıdır. Herhangi bir anda, bir cisim üzerindeki kuvveti hesaplamak için o anda diğer bütün cisimlerin konum ve hız bilgilerine gereksinim vardır.

Kullanılan zaman adımını sabit tutarak oluşan yakın çiftler için yeterli hassasiyette sayısal çözümler üretilmesi özellikle astrofiziksel çok cisim problemlerinde olası değildir. Herhangi bir anda oluşan farklı yakınlıktaki ikili, üçlü grupların farklı zaman adımlarıyla integre edilmeleri gerekebilir. Literatürdeki çalışmalar da zaten enerji korunumu üzerindeki hata birikimlerinin çoğunlukla oluşan yakın çiftlerden kaynaklandığını göstermektedir.

Çok-cisim problemlerinde birbirinden çok farklı ölçekte hızlara sahip cisimler aynı yapı içerisinde bulunabilirler. Yakın ve uzak çiftler için, oransal olarak hızlı ya da yavaş hareket eden cisimler için, değişken de olsa aynı zaman adımını kullanmak yeterli bir yaklaşım değildir. Aksi takdirde görece yavaş hareket eden uzak bileşenleri de en yakın çiftin gereksinim duyduğu en küçük zaman adımı ile integre etmemiz gerekir. Bu hem yuvarlama hatalarının sayısını yapılan işlem sayısındaki artma sebebiyle daha baskın hale getirir, hem de çok yüksek zaman tüketimine sebep olur.

Bu sebeplerle, değişken zaman adımı kullanmak tek başına yeterli değildir. Her cismin kendi gereksinim duyduğu zaman adımıyla integre edildiği ayrık zaman adımlı algoritmalar bu noktada öne çıkar (McMillan, 1986). Birbirine oldukça yakın ama tam olarak aynı olmayan zaman adımlarının bloklar halinde toparlandığı algo-

ritmalara da ayrık blok zaman adımları denir. Bu yaklaşım ayrık zaman adımlı algoritmaları daha yüksek başarımlı, daha uygulanabilir algoritmalar haline getirir. Blok zaman adımlı algoritmalar ayrıca çok-cisim algoritmalarının paralelleştirilme çalışmaları açısından da önemlidirler.

Her ne kadar çok-cisim probleminin çözümünde çeşitli yaklaşım yöntemleri kullanılabilir olsa da, bir çok durumda en yüksek doğruluk yine çok-cisim probleminin doğrudan integrasyonuyla sağlanmaktadır. Hatta büyük ölçekte yapılan benzetimler için kullanılan bazı yöntemlerde de küçük gruplar üzerinde yine doğrudan integrasyon tercih edilmektedir (Heggie ve Hut, 2003).

Son elli yılda yüksek enerji korunumu açısından öne çıkan başlıca yöntemler simplektik şemalar ve zaman-simetrik şemalardır. Her iki seçenek de integrasyon şemasında değişken zaman adımları kullanıldığında özelliklerini kaybetmektedirler. Literatürde değişken zaman adımı kullanıldığında bozulan zaman simetrisini tekrar sağlamak için önerilen başarılı bir şema vardır (Hut vd., 1995). Ancak bu şemayı doğrudan ayrık blok zaman adımlı şemaya uygulamak, zaman simetrisini sağlamak için yeterli değildir.

Bu çalışmada, zaman simetrik ayrık blok zaman adımlı bir algoritma geliştirmek için çalışılmıştır. Hem iki-cisim problemi analitik çözüme sahip olduğu için, hem de çok-cisim problemlerinde en büyük hata birikimleri yakın çiftlerden kaynaklandığı için, öncelikli olarak iki cisim problemi üzerinde durulmuştur. Blok zaman adımlı şemanın zaman simetrisini sağlayacak şema geliştirildikten sonra da, çok-cisim problemine uygulama yoluna gidilmiştir. Algoritmanın sahip olduğu yinelemeli yapı, ayrık blok zaman adımları kullanıldığında, her bir zaman adımı için geçmiş bilgilere gereksinim duymaktadır. Bu problemin üstesinden gelebilmek için çağ tabanlı yineleme kavramı geliştirilmiştir. Bu yineleme yapısında belirli alt zaman blokları içerisinde geçmiş bilgiler tutulur ve yineleme işlemleri bu aralıklar içerisinde yapılır. Çalışmanın devamında bu alt zaman aralıklarının değişken tutulması sağlanmıştır.

Sayısal testlere geçildiğinde, algoritma hem leapfrog, hem de 4, 6 ve 8. Dereceden Hermite integrasyon şemaları ile, uzun süreli integrasyon süreleriyle sınanmıştır (Makino ve Aarseth, 1992; Nitadori ve Makino, 2007).

### Zaman-simetrik blok zaman adımlı algoritma

Zaman simetrisine sahip ayrık blok zaman adımlı algoritmanın iki önemli ayağı vardır; çağ kavramı (era) ve zaman adımlarını simetrikleştirme şeması.

Çağ kavramı, yineleme işlemleri için gerekli bilgilerin bellekte tutulacağı alt zaman aralıkları için geliştirilmiştir. İntegrasyonun herhangi bir anında diğer bütün cisimlerin konum, hız, zaman, ivme ve gereken üst dereceden türev bilgileri bu zaman aralıkları için bellekte tutulur. Bütün cisimler çağ sonunda aynı zamana geldiklerinde geçmiş bilgiler silinir.

Blok zaman adımları  $n$ 'inci seviyede bir blok için eşitlik(1)'deki gibi tanımlanır:

$$\Delta t_n = \frac{\Delta t_1}{2^{n-1}} \quad (1)$$

Zaman-simetrik blok zaman adımları için kullandığımız ilk kriter, Hut ve diğerleri (1995) tarafından verilen şemadan hareketle, blok zaman adımı (1) tanımıyla birleştirilerek aşağıdaki gibi verilir:

$$n = \min_{k \geq 1} \left\{ k \mid \frac{1}{2^{k-1}} \leq \frac{(\delta t_i + \delta t_{i+1})}{2} \right\} \quad (2)$$

Burada  $\delta t_i$ , cismin  $i$ 'inci adımdaki değişken zaman adımı olmak üzere herhangi bir kritere göre belirlenebilir. Bu çalışmada eşitlik (3)'deki yaklaşım kullanılmıştır:

$$\delta t_i = \eta \min_j \frac{|r_{ij}|}{|v_{ij}|} \quad (3)$$

Eşitlik (3)'de  $\eta$  beklenen doğruluk mertebesi için bir parametre,  $r_{ij}$  ve  $v_{ij}$ ,  $i$ 'inci ve  $j$ 'inci ci-

simler arasındaki görece konum ve hız vektörlerinin büyüklükleridir.

Blok-zaman adımlarını simetrikleştirmek için Algoritma 1'deki adımlar izlenir.

---

#### Algoritma 1: Simetrikleştirme şeması

---

```

for  $i = 1$  to yineleme_sayısı do
    if  $time == odd\ time$  then
        if  $dt_i \neq \Delta t_n$  then
             $dt_i = dt_i / 2$ 
        end if
    end if
    if  $time == even\ time$  then
        if  $dt_i < \Delta t_n$  then
             $dt_i = dt_i * 2$ 
        end if
        if  $dt_i == \Delta t_n$  then
             $dt_i = \Delta t_n$ 
        end if
        if  $dt_i > \Delta t_n$  then
             $dt_i = dt_i / 2$ 
        end if
    end if
end for

```

---

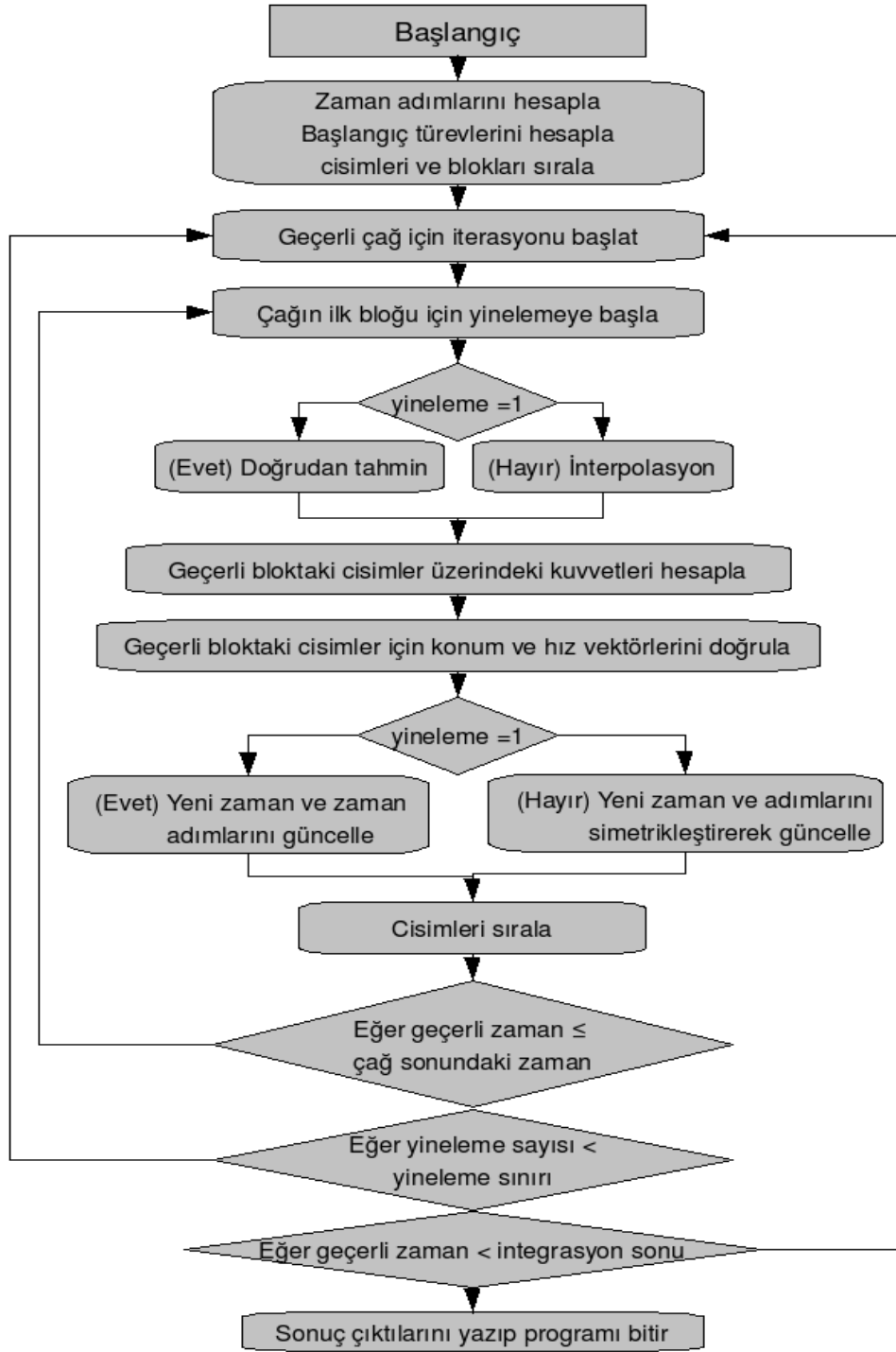
Burada, geçerli olan zaman, blok zaman adımının çift katıysa çift (*odd*), değilse tek (*odd*) zaman kabulü yapılır.

Alt zaman aralığından (era) ilk geçişte standart blok zaman adımlı şema, herhangi bir simetrikleştirme çabası olmaksızın kullanılır ve bütün cisimlerin gerekli adım bilgileri tutulur.

İlk iterasyon adımı olan ikinci geçişte, bellekte tutulan önceki adım bilgileri kullanılarak zaman-simetrik interpolasyon yapılır. Bu adımda, zaman adımları da Algoritma 1'deki şema kullanılarak simetrik hale getirilir. Bu adımdaki uygulama yineleme sayısınca tekrarlanır. Algoritmanın akış şeması Şekil 1'deki gibidir.

### İntegrasyon şemaları

Bu çalışmada, zaman-simetrik leapfrog, ve dördüncü dereceden Hermite ile, altıncı ve



Şekil 1. Akış şeması

sekizinci dereceden Hermite integrasyon şemaları kullanılmıştır. Sayısal testlerde farklı sayıda Plummer modeli başlangıç koşulları kullanılmıştır. İzin verilen en büyük zaman adımı ile alt zaman aralıklarının büyüklüğü eşit ve 1/64 olarak alınmıştır. Uzun süreli integrasyonlarda hata birikimini belirginleştirmek için

500-cisim problemlerinde doğruluk parametresi 0.5 olarak alınmıştır. Gerçekçi benzetimler için aslında bu çok büyük bir sayıdır. Ancak burada uzun süreli integrasyonda hata birikimlerini belirginleştirmek için özellikle kullanılmıştır. Birimler standart çok-cisim birimleri olarak alınmıştır.

### Leapfrog

Leapfrog, ikinci dereceden, zaman-simetrik bir integrasyon şemasıdır. İkinci dereceden olduğu halde uygulama kolaylığı ve zaman-simetrik yapısı sebebiyle oldukça etkili bir integrasyon yöntemi olarak sıklıkla kullanılır. Konum ve ivme vektörleri tam sayı adımlarla, hız vektörleri yarım adımlarla hesaplanır. Eşitlik (4)'deki gibi verilir:

$$\begin{aligned} r_{i+1} &= r_i + v_i \delta t + a_i \frac{\delta t^2}{2} \\ v_{i+1} &= v_i + (a_i + a_{i+1}) \frac{\delta t}{2} \end{aligned} \quad (4)$$

Burada  $r, v$  ve  $a$  sırasıyla konum, hız ve ivme vektörleri,  $i$  adım sayısı,  $\delta t$  zaman adımına karşılık gelir. İvme doğrudan konum vektöründen hesaplanır.

Sabit zaman adımları kullanıldığında toplam enerji üzerinde doğrusal olarak büyüyen bir hata birikimi görülmez.

Leapfrog integrasyonu ile 20 farklı Plummer 100-cisim problemine ilişkin sonuçlar Şekil 2'de gösterilmiştir.  $\Delta E$ , toplam enerji üzerindeki

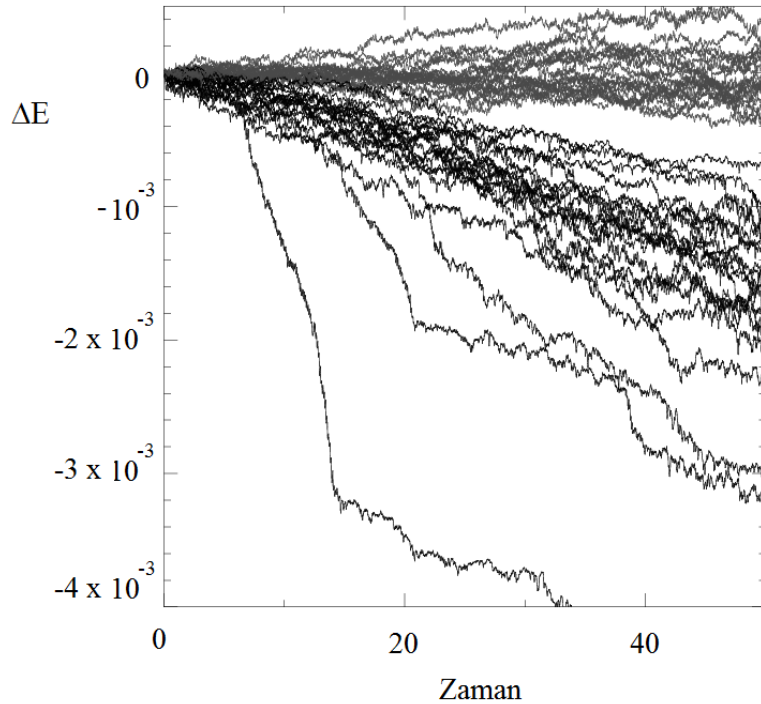
hataya karşılık gelmektedir. Burada gri olarak görülen eğriler beş yineleme ile, zaman-simetrik blok zamanlı algoritma kullanılarak alınan sonuçlardır. Doğrusal artış gösteren siyah eğriler ise klasik blok zaman adımli algoritma ile alınan sonuçlardır.

### 4. Dereceden Hermite

Dördüncü dereceden Hermite integrasyonu, çok-cisim integrasyon yöntemleri içerisinde önde gelen şemalardan biridir. Hermite interpolasyon polinomları kullanılarak üretilmiştir. Aynı dereceden bölünmüş fark denklemlerinden daha yüksek doğruluğa sahiptir. Bir tek adım için dördüncü dereceden Aarseth şemasından daha çok hesap gerektirdiği halde, iki kat daha büyük zaman adımlarıyla çalışabildiği için toplam hesap yükü daha düşüktür.

Tahmin adımları için aşağıdaki Taylor açılımları kullanılır;

$$\begin{aligned} r_{i+1} &= r_i + v_i \delta t + a_i \frac{(\delta t)^2}{2} + j_i \frac{(\delta t)^3}{6}, \\ v_{i+1} &= v_i + a_i \delta t + j_i \frac{(\delta t)^2}{2}. \end{aligned} \quad (5)$$



Şekil 2. 20 farklı 100 cisim problemi için leapfrog ile elde edilen sonuçlar

Burada  $j$ , ivmenin birinci türevidir (jerk). Doğrulama adımları da eşitlik (6)'daki gibi hesaplanır.

$$\begin{aligned} v_{i+1}^c &= v_i + (a_{i+1}^{new} + a_i) \frac{\delta t}{2} - (j_{i+1}^{new} - j_i) \frac{(\delta t)^2}{12}, \\ r_{i+1}^c &= r_i + (v_{i+1}^c + v_i) \frac{\delta t}{2} - (a_{i+1}^{new} - a_i) \frac{(\delta t)^2}{12}. \end{aligned} \quad (6)$$

Şekil 3'te dördüncü dereceden Hermite integrasyonu kullanılarak elde edilen sonuçlar görünmektedir. Burada gri tonda görülen ve büyük hata birikimine sahip olan sonuçlar yine zaman-simetrik olmayan blok zaman adımlı algoritmaya aittir. Zaman-simetrik blok zaman adımlı algoritma için ise 300 birim zaman kadar herhangi bir doğrusal hata artışı görülmemektedir.

## 6. Dereceden Hermite

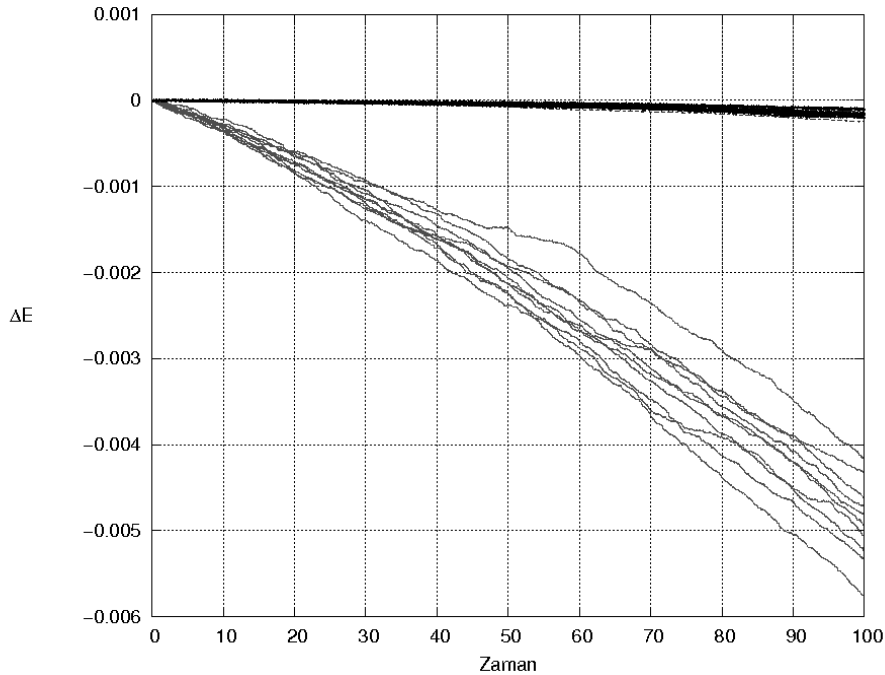
Altıncı dereceden Hermite integrasyonu, (Keigo ve Makino, 2007) yeni geliştirilmiş bir integrasyon yöntemidir. Dördüncü dereceden Hermite integrasyon şemasına göre daha üst dereceden türevlere gereksinim duyulduğu için, ivme vektörleri doğrudan hesaplanmazlar. Tahmin adımları için eşitlik (7)'deki Taylor açılımları kullanılır.

$$\begin{aligned} r_{i+1} &= r_i + v_i \delta t + a_i \frac{(\delta t)^2}{2} + j_i \frac{(\delta t)^3}{6} + s_i \frac{(\delta t)^4}{24}, \\ v_{i+1} &= v_i + a_i \delta t + j_i \frac{(\delta t)^2}{2} + s_i \frac{(\delta t)^3}{6}, \\ a_{i+1} &= a_i + j_i \delta t + s_i \frac{(\delta t)^2}{2}. \end{aligned} \quad (7)$$

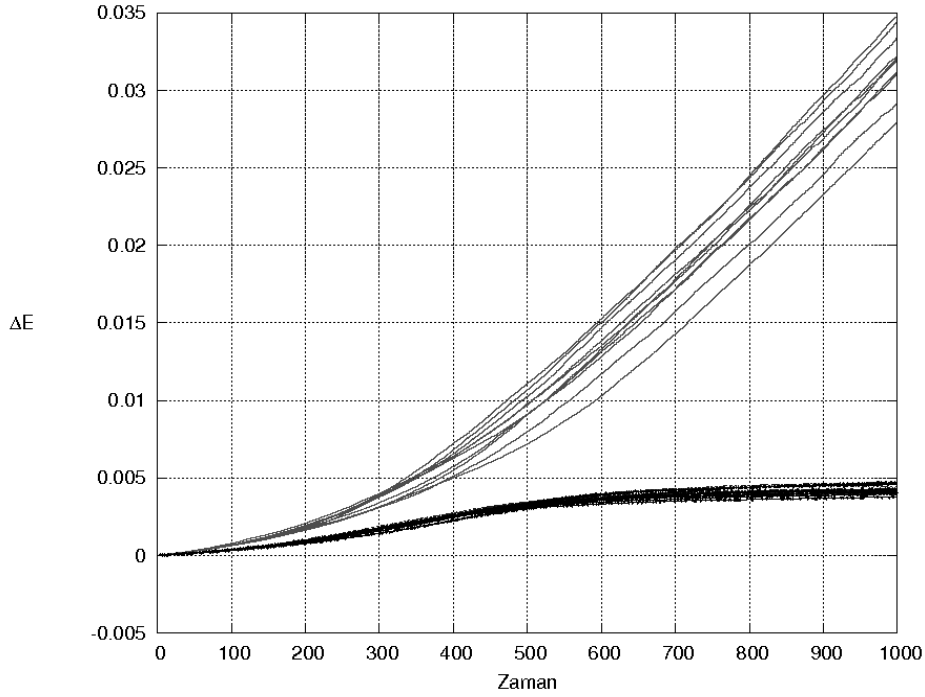
Burada  $s$ , ivmenin ikinci türevine karşılık gelir. Doğrulama adımları da eşitlik (8)'deki gibi hesaplanır;

$$\begin{aligned} v_{i+1}^c &= v_i + (a_{i+1}^{new} + a_i) \frac{\delta t}{2} - (j_{i+1}^{new} - j_i) \frac{(\delta t)^2}{10} \\ &\quad + (s_{i+1}^{new} - s_i) \frac{(\delta t)^3}{120}, \\ r_{i+1}^c &= r_i + (v_{i+1}^c + v_i) \frac{\delta t}{2} - (a_{i+1}^{new} - a_i) \frac{(\delta t)^2}{10} \\ &\quad + (j_{i+1}^{new} - j_i) \frac{(\delta t)^2}{120}. \end{aligned} \quad (8)$$

Şekil 4'te altıncı dereceden Hermite integrasyon şeması için alınan sonuçlar görünmektedir. Zaman-simetrik blok zaman adımı kullanıldığı durum siyah eğrilerle gösterilmiştir. Uzun süreli integrasyonda, normalden çok büyük alınan



Şekil 3. 10 farklı 500 cisim problemi için 4. dereceden Hermite ile elde edilen sonuçlar



Şekil 4. 10 farklı 500 cisim problemi için 6. dereceden Hermite ile elde edilen sonuçlar

doğruluk parametresiyle bile, algoritmanın etkinliğini koruduğu gözlenmektedir.

### 8. Dereceden Hermite

Sekizinci dereceden Hermite integrasyonu da altıncı dereceden Hermite gibi (Nitadori ve Makino, 2007) yeni geliştirilmiş bir integrasyon yöntemidir. Altıncı dereceden Hermite integrasyon şemasına göre daha üst dereceden türevlere gereksinim duyulur. Tahmin adımları için eşitlik (9)'daki Taylor açılımları kullanılır.

$$\begin{aligned}
 r_{i+1} &= r_i + v_i \delta t + a_i \frac{(\delta t)^2}{2} + j_i \frac{(\delta t)^3}{6} + s_i \frac{(\delta t)^4}{24} \\
 &\quad + c_i \frac{(\delta t)^4}{120}, \\
 v_{i+1} &= v_i + a_i \delta t + j_i \frac{(\delta t)^2}{2} + s_i \frac{(\delta t)^3}{6} + c_i \frac{(\delta t)^4}{24}, \quad (9) \\
 a_{i+1} &= a_i + j_i \delta t + s_i \frac{(\delta t)^2}{2} + c_i \frac{(\delta t)^3}{6}, \\
 j_{i+1} &= j_i + s_i \delta t + c_i \frac{(\delta t)^2}{2}.
 \end{aligned}$$

Benzer şekilde  $c$ , ivmenin üçüncü türevine karşılık gelir. Doğrulama adımları da eşitlik (10)'daki gibi hesaplanır.

$$\begin{aligned}
 v_{i+1}^c &= v_i + (a_{i+1}^{new} + a_i) \frac{\delta t}{2} - (j_{i+1}^{new} - j_i) \frac{3(\delta t)^2}{28} \\
 &\quad + (s_{i+1}^{new} + s_i) \frac{(\delta t)^3}{84} + (c_{i+1}^{new} - c_i) \frac{(\delta t)^4}{1680}, \\
 r_{i+1}^c &= r_i + (v_{i+1}^c + v_i) \frac{\delta t}{2} - (a_{i+1}^{new} - a_i) \frac{3(\delta t)^2}{28} \\
 &\quad + (j_{i+1}^{new} - j_i) \frac{(\delta t)^2}{120} + (s_{i+1}^{new} - s_i) \frac{(\delta t)^4}{1680}.
 \end{aligned} \quad (10)$$

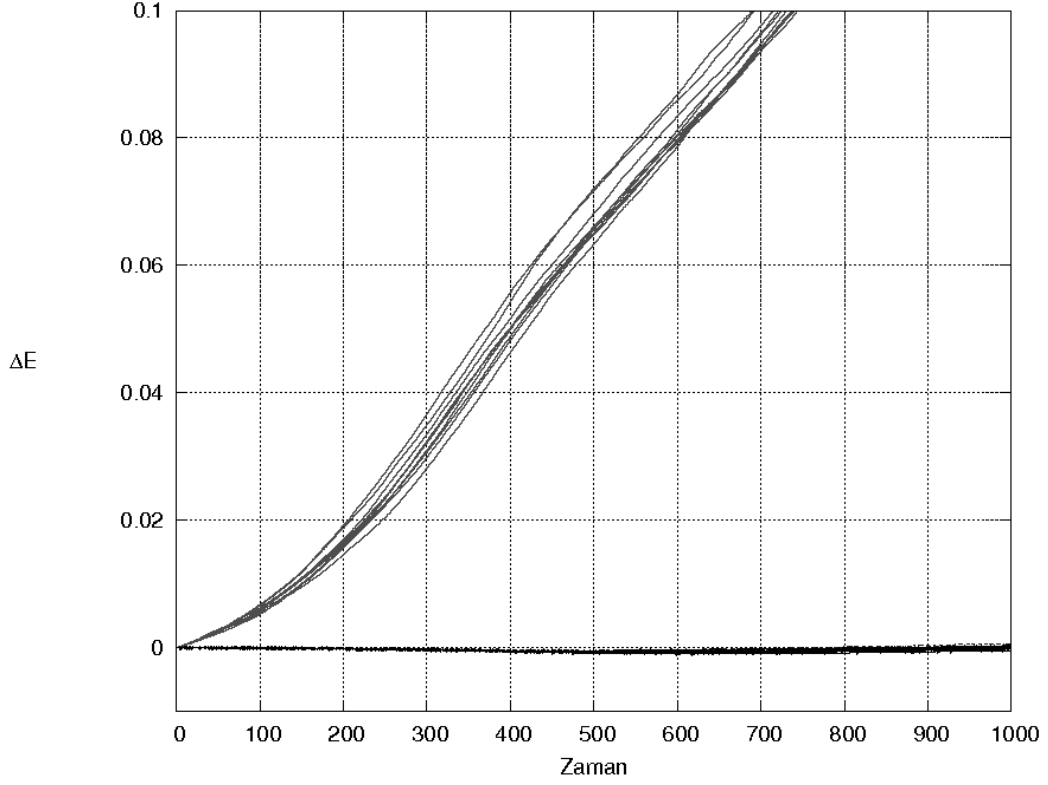
Şekil 5'te sekizinci dereceden Hermite ile de yine benzer sonuçlar alındığı görülmektedir.

### Değişken çağ tabanlı algoritma

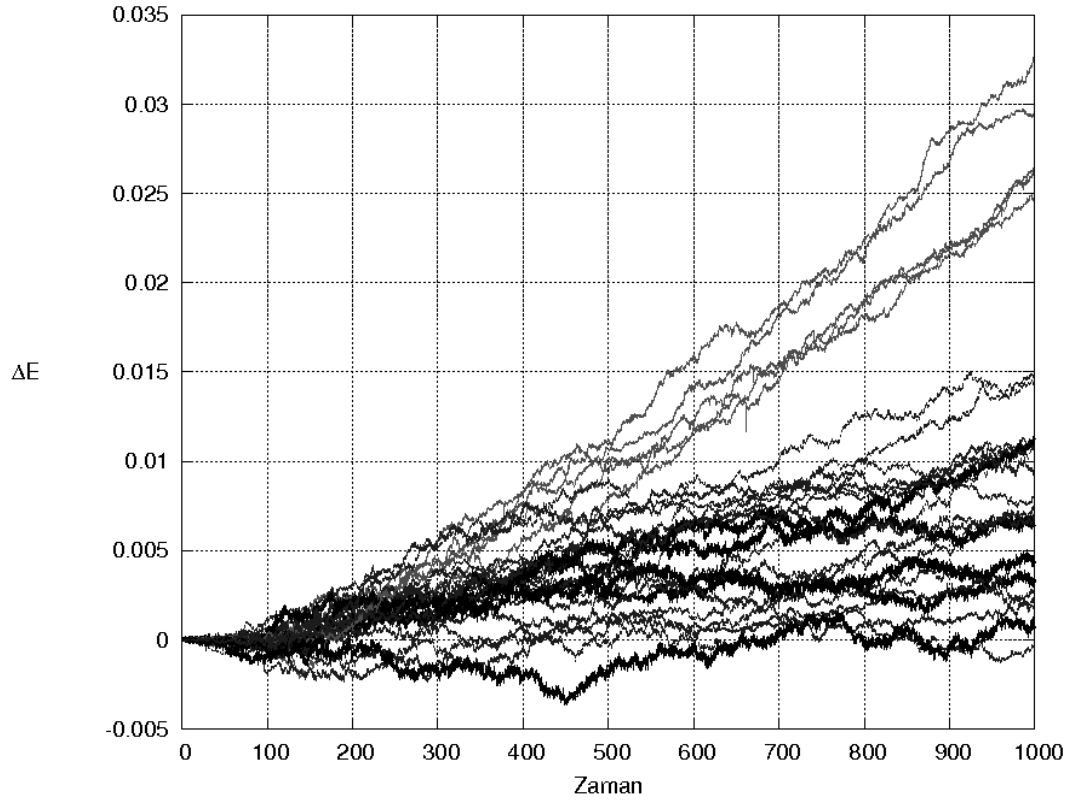
Algoritmanın önemli bir parçası olan çağ tabanlı yinelemede (era-based iteration) bu çağ aralıklarının büyüklüğünün nasıl seçileceği önemlidir. Çok büyük seçilmesi hem aşırı bellek tüketimine sebep olacağı, hem de gereksinim duyulan hesaplama süresini arttıracacağı için uygun olmaz. Ancak ne kadar küçük seçilebileceği düşünülmesi gereken bir noktadır. Çalışmanın devamında bu büyüklük üzerine detaylı sınamalar yapılmıştır.

Şekil 6'da zaman-simetrik blok zaman adımlı algoritma için yapılan testlerden biri





Şekil 5. 10 farklı 500 cisim problemi için 8. dereceden Hermite ile elde edilen sonuçlar



Şekil 6. 5 farklı 100 cisim problemi ve 5 farklı çağ aralığı için, zaman-simetrik blok zaman adımlı şema ve leapfrog integrasyonu ile alınan sonuçlar

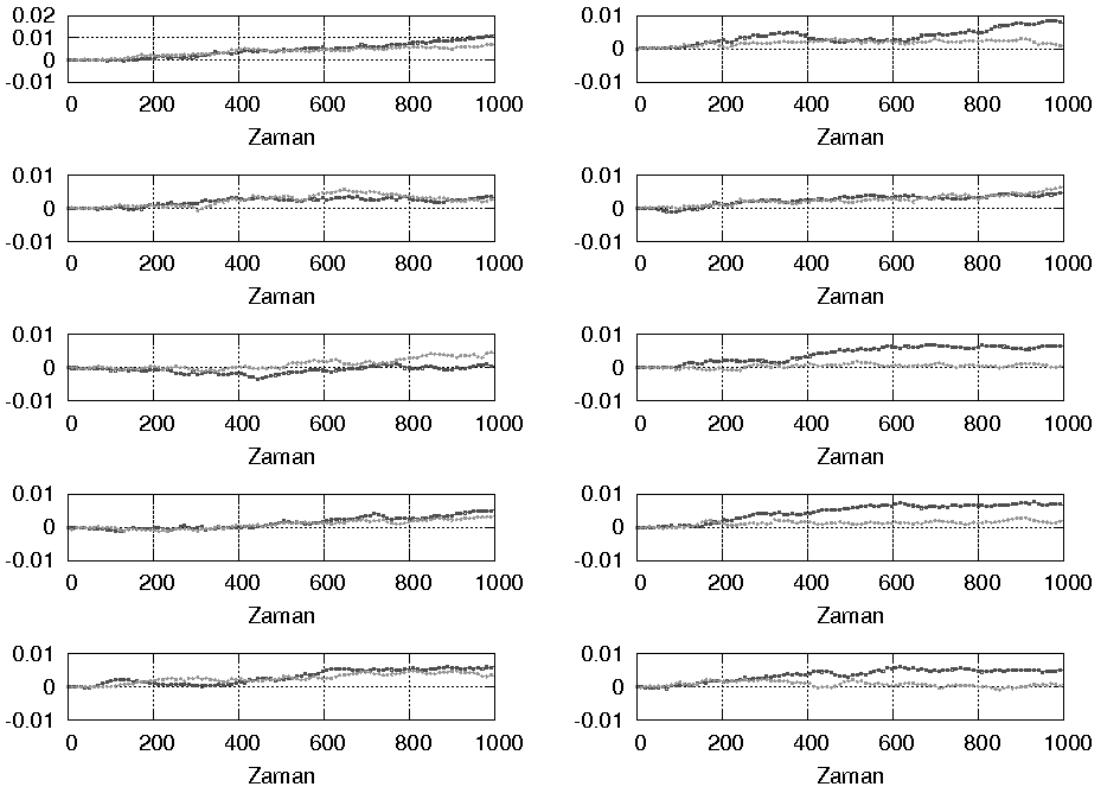
görülmektedir. Burada izin verilen en büyük zaman adımının beş farklı tam katı, çağ aralığı olarak seçilmiştir. Beş farklı 100-cisim problemi için her bir çağ aralığı kullanılarak sonuçlar alınmıştır. Bu aralıklar 0.015625, 0.03125, 0.0625, 0.125, ve 0.25'dir. Toplam enerji üzerindeki hataların en büyük olduğu durumlar en büyük çağ aralıklarına karşılık gelmektedir. Yineleme sayısı olarak 3 alınmıştır. Yineleme sayısı artırıldığında en büyük çağ aralığı için de daha iyi sonuçlar elde edilir. Ancak hem bellek tüketimi hem de integrasyon süreleri artar.

Çağ aralığı için alınabilecek en basit değer aralığı olarak 1 birim zaman ile izin verilen en büyük zaman adımı aralığı görülmektedir. Burada önerilen, en büyük zaman bloğu ile en küçük bloğun zamanı arasındaki farkı kullanmaktır. Bu değerın çok büyük ya da izin verilen en büyük zaman adımından küçük olması durumları da kontrol edilebilir. Algoritma değişken çağ aralığı tabanlı yinelemelere uygun hale getirilerek çeşitli testler yapılmıştır.

Şekil 7'de 10 farklı 100-cisim problemi için karşılaştırmalı sonuçlar görülmektedir. Değişken çağ aralığı tabanlı yineleme ile karşılaştırmak için sabit çağ aralığı değeri olarak, önceki testlerde en iyi sonuç veren büyüklük, yani izin verilen en büyük zaman adımı seçilmiştir. Değişken çağ tabanlı yineleme, genellikle daha iyi sonuç vermekle beraber, en kötü sonuç bile izin verilen en büyük zaman adımı ile aynı mertebede sonuçlar vermektedir.

## Sonuçlar

Bu çalışmada, çok-cisim problemleri için zaman-simetrik ayırık blok adımli bir algoritma geliştirilmiştir. Algoritmanın etkinliği iyi bilinen zaman-simetrik leapfrog integrasyon şeması ile olduğu gibi, 4, 6 ve 8inci dereceden Hermite integrasyon şemaları ile de sınanmıştır. Her durumda da ayırık blok-zaman adımli algoritma ile oldukça iyi sonuçlar elde edilmiştir. Çalışmanın son safhasında algoritma değişken çağ aralıklı yinelemelere de olanak sağlayacak şekilde genelleştirilmiştir.



Şekil 7. 10 farklı 100 cisim problemi için değişken ve sabit çağ aralığı tabanlı yinelemelerle alınan sonuçlar (Burada zaman-simetrik blok zaman adımli şema leapfrog integrasyonu ile kullanılmış, yineleme sayısı olarak da 3 alınmıştır)

## Kaynaklar

- Aarseth, S., (2003). *Gravitational N-body simulations*, Cambridge University Press.
- Heggie, D. ve Hut, P., (2003). *The gravitational million-body problem*, Cambridge University Press.
- Hut, P., Makino, J. ve Mcmillan, S., (2003). Building a better leapfrog, *Astrophysical Journal*, **415**, 715.
- Makino, J. ve Aarseth, S., (1992). On a Hermite integrator with A-C scheme for gravitational many-body problems, *Publications Astronomical Society of Japan*, **44**, 2,141-151.
- Mcmillan, S., (1986). The vectorization of small N-integrators, *The Use of Supercomputers in Stellar Dynamics*, **156**, 61.
- Nitadori, K. ve Makino, J., (2007). 6<sup>th</sup> and 8<sup>th</sup> order Hermite integrator for N-Body simulations, *New Astronomy*, **13**, 498-507.