

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SIVI YAPI İNTEGRAL DENKLEMLERİNİ
ÇÖZMEK İÇİN BİR YÖNTEM

34660

Elif TEKİN

F.B.E. Matematik Anabilim Dalında
hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof. Dr. Behiç ÇAĞAL

T.C. YÜKSEKÖĞRETİM KURULU
TEZ KURULU

İSTANBUL, 1994

İÇİNDEKİLER

TEŞEKKÜR

ÖZET i

SUMMARY ii

GİRİŞ 1

1. PİCARD VE NEWTON-RAPHSON YÖNTEMLERİNİN KOMBİNASYONU 5

2. TEKNİK DETAYLAR 12

2.1 İntegral Denklemin Davranışı 12

2.2 Temel Fonksiyonların Seçimi 14

2.3 Jacobian' ın Hesaplanması 17

3. METODUN İNCELENMESİ 19

4. SONUÇ 23

EK.1. NEWTON-RAPHSON YÖNTEMİ 24

EK.2. BİLGİSAYAR PROGRAMI

KAYNAKLAR

ÖZGEÇMİŞ

TEŞEKKÜR

Tezimin danışmanlığını üstlenen hocam Prof. Dr. Behiç ÇAĞAL' a teşekkür ederim.

Ayrıca tez çalışmamda Y.T.Ü. Fizik ve Matematik Bölümü'ndeki tüm arkadaşlara yardımları için teşekkür ederim.

Ö Z E T

Bu çalışmada sıvı yapı için Percus - Yevick ve hypernetted chain denklemlerinin sayısal çözümlerini bulmak için bir yöntem anlatılacaktır. Hızla yakınsayan ve oldukça kararlı olan bu yöntem, Newton-Raphson ve geleneksel iteratif yöntemin bir karışımıdır. Tipik potansiyeller için sayısal testlerle bu yöntemin çok yüksek yoğunluklarda bile 20 veya 30 iterasyonda iyi yakınsayan çözümler verdiğini görülmüştür. Çok doğru olmasa bile ilk değerinin seçimi için ihtiyaç duyulan iterasyon sayısında hassaslığa gerek yoktur.

S U M M A R Y

In the work, a method of obtaining numerical solutions is represented to the Percus - Yevick and hypernetted chain equations for liquid structure. The method, which is rapidly convergent and very stable, is hybrid of the traditional iterative scheme and the Newton-Raphson technique. We show by numerical tests for typical potentials that the method gives well-convergent solutions in 20 or 30 iterations even for very high densities. The number of iterations needed is insensitive to the choice of initial estimate, even if this is extremely inaccurate.

GİRİŞ

Sıvılar gibi düzensiz yapıların mikroskobik ölçekte incelenmesi oldukça zordur. Çünkü bu sistemler katılarda olduğu gibi belirli bir yapıya sahip değildirler. Bu nedenle sıvı yapıyı oluşturan elemanlar (atomlar, iyonlar, moleküller v.s.) arasındaki etkileşmelerden yola çıkarak sıvıların yapısını verecek bağıntılar oluşturmak gerekir. Bunlardan en yaygın olarak kullanılan çift dağılım fonksiyonları ve radyal dağılım fonksiyonlarıdır. Bu dağılım fonksiyonları deneysel ve kuramsal olarak elde edilebilirler. Kuramsal çalışmalar; tamamen sayısal olan Monte-Karlo simülasyon tekniği, moleküler dinamik tekniği ve bu çalışmada incelenen integral denklem tekniğidir [Hansen et al, 1986].

İntegral denklem tekniğinde sistemi oluşturan parçacıkların birbiri ile etkileşmesinden kaynaklanan korelasyon fonksiyonlarına bakılır. Buna göre bir parçacık gözönüne alındığında, sistemin geri kalan kısmı ile oluşturduğu toplam korelasyonu, parçacıklarla tek tek etkileşmesinden kaynaklanan doğrudan korelasyon ve parçacıkların toplam etkileşmelerinden kaynaklanan dolaylı korelasyon toplamı olarak tanımlanabilir. Böylece $h(r)$ toplam korelasyon fonksiyonu,

$$h(r) = c(r) + n \int dr' c(|r - r'|) h(r') \quad (1)$$

şeklinde yazılır. Burada sağ taraftaki ilk terim doğrudan korelasyon ikinci terim ise dolaylı korelasyonu verir. (1) denklemi Ornstein-Zernike integral denklemi olarak bilinir. Toplam korelasyon fonksiyonu $h(r)$ aynı zamanda

$$h(r) = g(r) - 1 \quad (2)$$

eklinde de ifade edilir. Burada $g(r)$ çift dağılım fonksiyonudur. Orjinde seçilen bir parçacıktan r kadar uzaklıkta dr küresel kabuk içindeki parçacıkların sayısı n_0 ise

$$n_0 = 4\pi r^2 n g(r) dr \quad (3)$$

azılabilir. Burada $n = N/V$ sayı yoğunluğu, (N: Toplam parçacık sayısı, V: Sistemin hacmi olmak üzere) $r \rightarrow 0$ için $g(r) \rightarrow 0$, $r \rightarrow \infty$ için $g(r) \rightarrow 1$ asimtotik değerine sahiptir. Aynı zamanda $4\pi r^2 g(r)$ büyüklüğü ise radyal dağılım fonksiyonu olarak adlandırılır. Sıvılar gibi düzensiz yapıların incelenmesinde oldukça önemli olan bu çift dağılım fonksiyonunun elde edilebilmesi için (1) integral denklemini çözmek gerekir. Kuramsal olarak parçacıklar arasındaki bütün etkileşmeler gözönüne alınarak $h(r)$ elde edilebilir. Fakat pratik olarak sistemdeki parçacık sayısı çok fazla ise bütün etkileşmeleri gözönünde bulundurmak problemi çözümü elde edilemeyecek duruma getirebilir. Bu nedenle sistemdeki parçacıklar arasında belli türden etkileşmeler gözönüne alınarak bazı yaklaşımlar yapılır. Bu yaklaşımlar Percus - Yevick ve hypernetted chain yaklaşımlarıdır [Allen et al,1987].

Bu çalışmada Percus - Yevick (PY) ve hypernetted chain (HNC) denklemlerinin sayısal çözümlerini elde etmek için bir yöntem anlatılacaktır. Yöntem karardır, hızla yakınsar. Ayrıca bilgisayarda işlem zamanı da kısadır. Bu çalışmanın yapılmasının iki nedeni vardır:

(a) Sıvı yapı integral denklemlerinin sayısal çözümlerine hala sık sık gerek duyulmaktadır. Geleneksel iteratif yöntemle bunu elde etmek bıkıtırıcı olmaktadır. Bu yüzden genellikle daha iyi doğrulukla sonuç veren bir yaklaşım gerektirir ve o zaman bir kaç yüz iterasyon gerekli olabilir.

(b) Fizikte integral denklemler sık sık ortaya çıktığından geliştirilen tekniğin uy-

ulama alanı fazladır.

$\phi(r)$ potansiyelinden dolayı etkileşen tek bileşenli bir sistem için PY ve HNC denklemleri aşağıdaki gibi yazılabilir [Watts,1973]:

$$c(r) = (1 + \gamma(r))(exp(-\beta\phi(r)) - 1) \quad (PY) \quad (4a)$$

$$c(r) = exp(-\beta\phi(r) + \gamma(r)) - \gamma(r) - 1 \quad (HNC) \quad (4b)$$

burada

$$\gamma(r) = h(r) - c(r)$$

şudur. $h(r)$ bilindiği gibi toplam korelasyon fonksiyonu ve β bilinen sabittir. Sayısal hesaplamalarda, $|r|$ nin büyüklüğü genellikle $r_i = (i - 1)\delta r$ örgü(mesh) noktalarına karşılık gelen N tane noktaya bölünür ve $\gamma(r)$ fonksiyonları bu noktalardaki $\gamma_i = \gamma(r_i)$ değerleri ile gösterilir.

Bu denklemlerin geleneksel çözüm yönteminde [Watts, 1973] şu yol izlenir: Fonksiyonların biri için bir başlangıç yaklaşımı olarak $\gamma(r)$ alınır ve bu $c(r)$ yi elde etmek için denklem (6) da yerine konur. Denklem (1), $c(r)$ cinsinden $\gamma(r)$ yi verir. Bu da yeni bir $\gamma(r)$ yaklaşımını elde etmek için kullanılır. Eğer olanaklı ise çevrim yakınsamaya kadar devam eder. Bu yöntem sayısal analizde [Oretega et al,1970] Picard Yöntemi olarak adlandırılan yöntemin bir çeşitidir. Bu isimden dolayı yönteme Picard yöntemi diyeceğiz.

İyi bir başlangıç yaklaşımı elde edilebilse bile yukarıdaki işlemde genellikle yakınsama yine de yavaştır. Bu, erimiş tuzlar için HNC denklemi üzerinde yapılan çalışmalarla çok iyi gösterilmiştir. Hansen ve McDonald (1975) yoğun erimiş bir tuz için çözüm bulmada toplam 4000 iterasyona gereksinim duymuşlardır. Adams (1976) ve Larsen (1978) de çalışmalarında doğruluğu kabul edilen başlangıç yaklaşımı seçerek bir kaç yüz iterasyon kullanmışlardır.

(1) ve (4) denklemleri N tane lineer olmayan eş zamanlı bir denklem kümesi olarak kabul edilebilir. Başka bir deyişle, denklemlerin kümeleri çoğunlukla çok üçlü NR yöntemi kullanılarak çözülürler [Ortega et al,1970-Ostrowski,1966]. Varyayalım ki; N tane $\{x_j\}$ değişkenli N tane $f_i(\{x_j\})$ fonksiyonlarının bir kümesine sahip olunsun ve f_i yi sıfır yapan $\{x_j^*\}$ değeri bulunmak istensin

$$f_i(\{x_j\}) = 0 \quad (5)$$

NR yöntemine göre $\{x'_j\}$ yeni yaklaşımı, $\{x_j\}$ eski yaklaşımından bulunur ve aşağıdaki gibi yazılır:

$$x'_i = x_i - \sum_j (J^{-1})_{ij} f_j(\{x_i\}) \quad (6)$$

Buradaki J_{ij} jacobian matrisi ise şu şekilde bulunur:

$$J_{ij} = \frac{\partial f_i}{\partial x_j} \quad (7)$$

$\{x_j^*\}$ ye başarılı yaklaşımlar elde etmek için (7) denklemi peş peşe işleme sokulur.(EK.1. de yöntem daha ayrıntılı ele alınmıştır.)

NR yönteminin güçlülüğü sonderece hızlı yakınsama özelliğinde yatmaktadır [Ortega et al,1970-Ostrowski,1966]. Bunun her bir iterasyondaki en büyük engeli, N boyutlu bir matrisin inversini hesaplama gereksinimidir. Kuşkusuz yöntemin dikkat gerektirdiği açıktır.

Geliştirilmek istenen düşünce, Picard yöntemiyle NR yöntemini birleştirerek bunu işletmedeki zorluğu yenmektir. Bu öyle bir yolla yapılabilir ki; NR yönteminin doğruluğu daha çok arttırılırken, Picard yönteminin yavaşlığı ve kararsızlığı giderilir. Bilgisayar sonuçlarının hızlı ve kararlı olduğu görülür. Yüksek yoğunluklarda bile çok kötü bir ilk yaklaşımla başlanarak, yaklaşık 20-30 iterasyonda

gerçek çözüm bulunabilir. Bilgisayar zamanı bir kaç saniyeden fazla değildir.



1. PİCARD VE NEWTON-RAPHSON YÖNTEMLERİNİN KOMBİNASYONU

1) $c(r)$ yi elde etmek için başlangıç yaklaşımı $\gamma(r)$ alınır ve (4) denklemi kullanılır.

2) $c(r)$ nin üç boyutlu fourier transformasyonu alınır ve γ nın fourier transformasyonunu elde etmek için

$$\tilde{\gamma}(k) = \frac{n\tilde{c}(k)^2}{(1 - n\tilde{c}(k))} \quad (8)$$

formunda yazılan OZ denkleminde yerine konur.

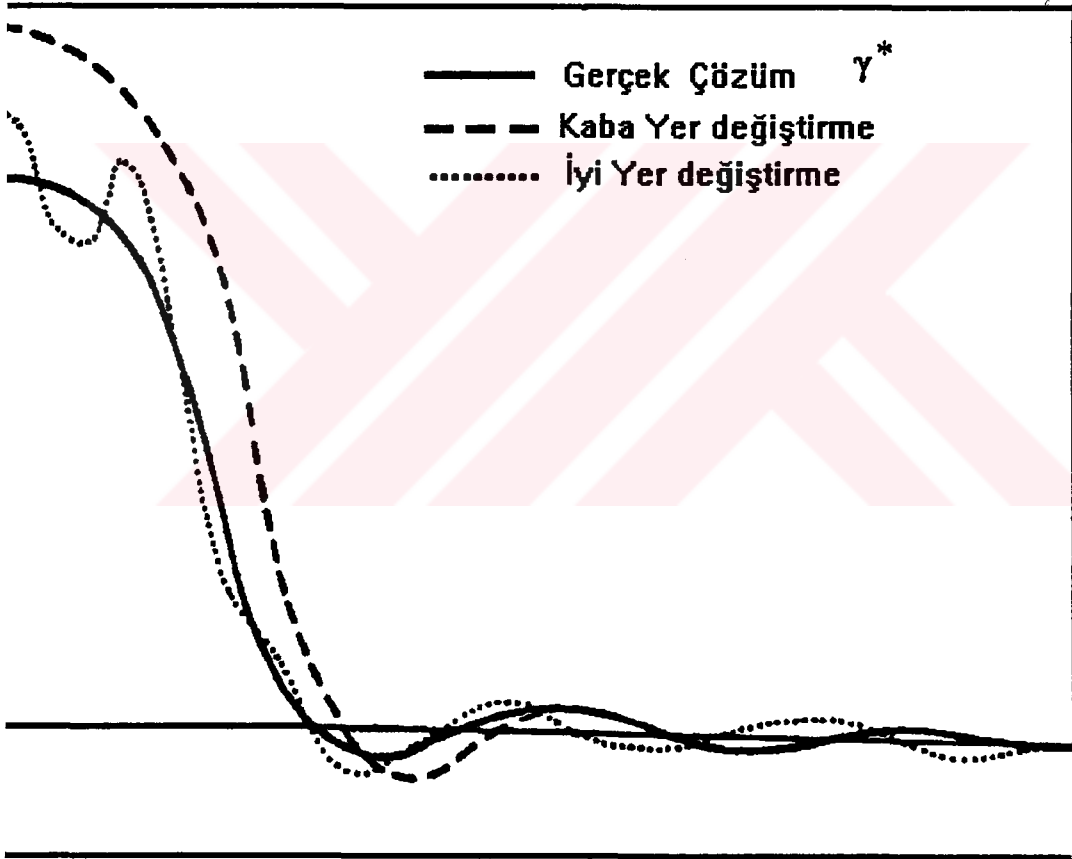
3) $\gamma'(r)$ çıkış yaklaşımını elde etmek için ters fourier dönüşümü alınır. Bu dönüşümler aşağıdaki gibi tanımlanmıştır.

$$\tilde{c}(k) = \int dr \exp(ik.r) c(r), \quad \tilde{\gamma}(k) = \int dr \exp(ik.r) \gamma(r) \quad (9)$$

Pratikte, kesikli fonksiyonların gösterimi $\{x_i\}$ gibidir ve genellikle onların bu formda kullanılmaları tercih edilecektir.

Gerçek çözümün olduğunu varsayalım ve bunun hakkında $\gamma_i = \gamma_i^* + \delta\gamma_i$ dönüşümünü düşünelim. Eğer γ_i ye başarılı elemanter döngüler uygulanması onları gerçek çözüme yönlitiyorsa, Picard yöntemi kararlıdır. $\delta\gamma_i$ çok küçük olmadıkça doğruluğun genel olmadığı gösterilebilir. Ayrıca değişik türde $\delta\gamma_i$ yer değiştirmeleri üzerinde de düşünülecektir. Özellikle kaba ve iyi yer değiştirmelerinin arasında ayırım yapılması gerekir. Kaba bir yer değiştirmeye γ_i nin tamamen biçimi veya büyüklüğünün değiştirilmesi kastedilir. Örneğin: $\delta\gamma_i = \alpha\gamma_i^*$ (burada α bir

ıdır). İyi bir yer deęiřtirme, γ_i^* gerek özümünün tanımlanan öleğinden ok daha küçük olan bir uzunluk öleğidir (yani fourier bileřenleri esas olarak uzun alga vektörleridir) veya bu sadece büyük r_i lere sahip olan γ_i nin kuyruğunu etkiler. Şekil-1 de yer deęiřtirmenin iki tipinin örnekleri verilmiştir.



ŞEKİL-1

Bizim amacımız için iyi yer değiştirmelerin en önemli özelliği, γ'_i çıkış değerlerinin γ_i giriş değerlerinin iyi yer değiştirmelerine duyarsız olmasıdır. Örneğin; (4) denklemlerinden açıkça anlaşılacağı gibi potansiyelin genişliği ötesinde büyük r_i ler için γ_i lerdeki bir değişim c_i üzerinde küçük bir etkiye sahiptir. Böylece değişim γ'_i üzerinde küçük bir etkiye sahip olacaktır. Benzer olarak, yüksek dalga vektöründe, γ_i deki bir değişim c_i de bir değişim oluşturacaktır. Bu bölgede (8) denkleminde dolayı $\tilde{\gamma}(k)$, $\tilde{c}(k)^2$ ye orantılı olduğundan bu değişim küçük bir etkiye sahip olacaktır.

İyi yer değiştirmeden daha çok kaba yer değiştirmeden doğan problem tipi için Picard yönteminin kararsızlığı, yöntemimizin anahtarını oluşturur. Yani yukarıdaki ile şu ifade edilmektedir. γ_i nin iyi kısmındaki değişimler γ'_i üzerinde küçük bir etkiye sahipse, o zaman gerçek çözümde uzakta olan iyi kısımdaki bir yer değiştirme, temel döngülerin ard arda işlemesiyle sıfıra indirgenecektir.

Eğer bu doğruysa, NR yöntemiyle γ_i nin kaba kısmının hesaplanmasını düşünmek doğaldır ve bırakılan iyi kısım Picard yöntemi ile halledilir. Büyük boyutlarda problem NR yöntemiyle sıkıntılı hal alır ve problemde şu değişiklikler olur: Kaba kısım kabalığından dolayı parameterelerin küçük bir sayısı ile kesinlikle belirtilebilir. Öyle ki; bu da küçük boyutta bir problemidir.

Bu fikri geliştirmek için, kaba ve iyi kısımlardan γ_i yi ayırtıran kesin bir yola ihtiyaç vardır. Bunu bir N-boyutlu γ vektörünün γ_i bileşenleri gibi yazabiliriz. Kaba ve iyi kısımlar ortogonal iki alt uzayda γ izdüşümleri gibi tanımlanırlar. Temel vektörlerin bir küçük ν sayısı ile gerilen kaba alt uzayın bileşenlerini $P_\alpha^i (x = 1, 2, 3 \dots \nu)$ olarak alacağız. γ nın ayrışmasını (yani bileşenlerini) şöyle

yazabiliriz:

$$\gamma_i = \sum_{\alpha} a_{\alpha} P_{\alpha}^i + \Delta\gamma_i \quad (10)$$

burada

$$\sum_i P_{\alpha}^i \Delta\gamma_i = 0 \quad \forall \alpha \quad (11)$$

dır. $\Delta\gamma_i$, γ nın iyi bilişeniyle türetebilir. Temel vektörler P_{α}^i , γ_i nin büyüklüğü ve kapsamlı biçiminin uygun bir gösterimini verecek şekilde seçilmelidir.

Şimdi NR yönteminin uygulanmasına geçelim. Varsayalım ki, (10) denklemiyle verilen γ ya etki eden bir elemanter döngü ile γ' oluşturulsun.

$$\gamma'_i = \sum_{\alpha} a'_{\alpha} P_{\alpha}^i + \Delta\gamma'_i \quad (12)$$

Eğer gerçek çözüm γ ise, şöyle yazılabilir:

$$d_{\alpha} \equiv a_{\alpha} - a'_{\alpha} = 0 \quad (13)$$

$$\Delta\gamma'_i = \Delta\gamma_i \quad (14)$$

Keyfi seçilen bir $\Delta\gamma_i$ genelde denklem (14) ü sağlamayacaktır. Biz (14) ü sağlayan a_{α} nın kümesine başvurabiliriz. Böyle bir küme var olduğu varsayalım, bunu bulmak için NR yöntemini kullanabiliriz. NR ile eski a_{α} yaklaşımından yeni bir $\bar{a}_{\alpha}$ yaklaşımı elde edilebilir:

$$\bar{a}_{\alpha} = a_{\alpha} - \sum_{\beta} (J^{-1})_{\alpha\beta} d_{\beta} \quad (15)$$

burada d_{β} , (13) denklemiyle tanımlanan farklar ve jacobian matris

$$J_{\alpha\beta} = \frac{\partial d_\alpha}{\partial a_\beta} \quad (16)$$

eski yaklaşım $\{a_\alpha\}$ daki değeriyle hesaplanmıştır, sabit $a_{\gamma(\neq\beta)}$ ve sabit $\Delta\gamma_i$ de türevlenebilir. Başarılı bir elemanter döngüyü gerçekleştirme ve a_α nın değiştirilmiş bir kümesini elde etmek için (15) denklemindeki d_α sonucunu kullanma bir NR döngüsü gibi işlenir. Eğer ardışık NR döngüleri yakınsarsa, $\Delta\gamma_i$ sabit tutularak, a_α^* olarak gösterilen a_α ları bularak, aşağıdaki döngünün işlemesiyle

$$\gamma_i^* = \sum_{\alpha} a_\alpha^* P_\alpha^i + \Delta\gamma_i \quad (17a)$$

şu özellik elde edilir:

$$\gamma_i'^* = \sum_{\alpha} a_\alpha^* P_\alpha^i + \Delta'\gamma_i. \quad (17b)$$

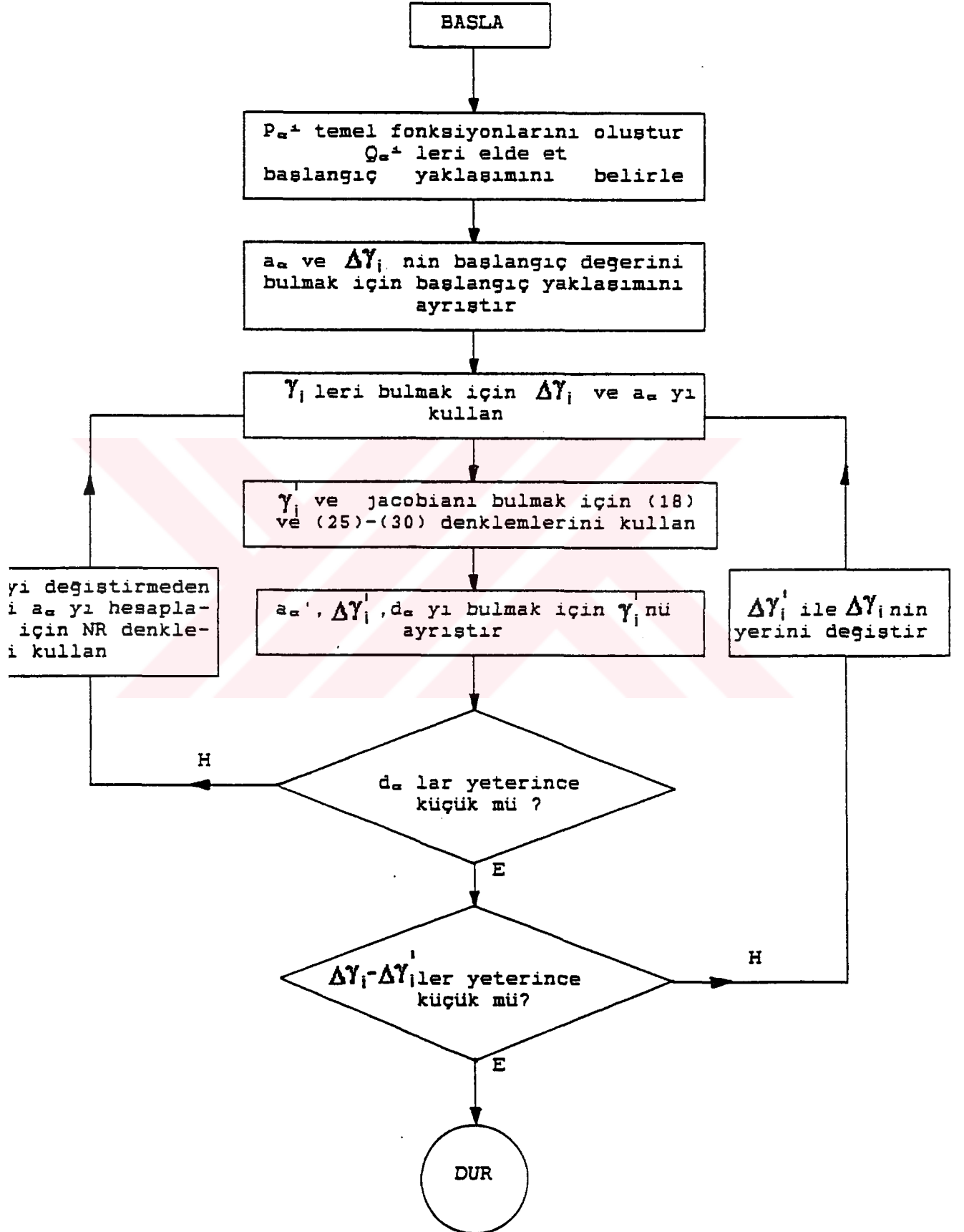
Bu da değişmez kaba kısımdır.

Ardışık işlemler sonucunda verilen bir $\Delta\gamma_i$ iyi kısmından (17) denklemlerine başka bir $\Delta\gamma_i'$ yaklaşımını bulabiliriz. NR döngülerinin bir kümesine iyileştirme döngüsü olarak da bakılabilir.

Şimdi konumuz (13) denklemi kadar (14) denklemini iyi sağlayacak γ yı bulmaktır. Bu, γ nın tamamen yakınsayana kadar iyileştirme döngüleri tekrarlanarak başarılabılır. Her bir iyileştirme döngüsünde $\Delta\gamma_i'$ iyi kısmı bir sonraki döngüde $\Delta\gamma_i$ iyi kısmı olarak alınır. Tüm hesaplama böylece iyileştirme döngülerinden oluşur. Burada bir iteratif adım, bir iyileştirme döngüsü olan Picard yöntemi ile $\Delta\gamma_i$ hesaplanması olarak kabul edilir. Tüm hesaplama Şekil-2 deki akış diyagramıyla özetlenmiştir.

Bu taslağın kullanışlı olabilmesi için çok emek harcamadan $J_{\alpha\beta}$ jacobian matrisini hesaplayabilmeliyiz. Bunu sağlayabilmek için kaba alt uzayın ν boyutu azaltılmalıdır. Böylece J nin kolayca inversi alınabilir.

ŞEKİL-2. AKIŞ DİYAGRAMI



2. TEKNİK DETAYLAR

2.1. İntegral Denklemin Davranışı

(1) denklemindeki geleneksel integral denklemi direkt integrasyonundan daha ok fourier transformu ile hesaplama tercih edilecektir. Çünkü dirket yöntemde şlemlerin sayısı N^2 ile orantılı olmasına rağmen fast fourier teknikleri ile ilk yön-em sadece $N \log_2 N$ işlem gerektirir. Bu büyük avantajından dolayı doğal olarak ransform yöntemin seçmek mantıklıdır.

PY ve HNC denklemleri kesikli formda şöyle yazılabilir:

$$c_i = (1 + \gamma_i)(\exp(-\beta\phi_i) - 1) \quad (PY) \quad (18a)$$

$$c_i = \exp(-\beta\phi_i + \gamma_i) - \gamma_i - 1 \quad (HNC) \quad (18a)$$

$$\tilde{c}_j = (4\pi\delta r/k_j) \sum_{i=1}^{1/2N-1} r_i \sin \mathbf{k}_j \mathbf{r}_i c_i \quad (18b)$$

$$\tilde{\gamma}_j = n\tilde{c}_j^2 / (1 - n\tilde{c}_j) \quad (18c)$$

$$\gamma'_i = (\delta k / 2\pi^2 r_i) \sum_{j=1}^{1/2N-1} k_j \sin \mathbf{k}_j \mathbf{r}_i \tilde{\gamma}_j \quad (18d)$$

burada

$$r_i = i\delta r, \quad k_j = 2\pi j / N\delta r \quad (19)$$

dir.

(18) denklemleri, γ_i üzerinde bir elemanter döngüyü işlemek için gereken işlemleri kesinlikle gerçekleştirir. Gerçekte c_i ve h_i den çok γ_i üzerinde bunu işlemek için zorlama yoktur. Bununla birlikte r_i nin bir fonksiyonu olarak γ_i nin düzlüğü mantıklı bir seçimi sağlar: Düzgün bir fonksiyon kolaylıkla bir kaç temel fonksiyonun terimlerinde iyi kısım küçülecek şekilde gösterilebilir. Böylece daha çok ağırlık etkin olan NR yöntemine verilecektir. c_i ve h_i fonksiyonları kabuk(core) apı civarında hızlı bir şekilde değiştiğinden, daha az uygundur.

2.2. Temel Fonksiyonların Seçimi

Temel fonksiyonların mümkün olan şekli sonsuzdur. Bir seçimimiz de, uygun, seçimi tek olmayan çatı fonksiyonlarının benzeş olan kesikli fonksiyonları olarak alınması olabilir. N hariç, r_i örgü noktalarından, kesikli ν tane noktadan oluşan bir alt küme seçelim. Bu kümenin elemanları $i_\alpha (\alpha = 1, 2, \dots, \nu)$ larla etiketlenmiştir. Burada $\alpha > \beta$ ise $i_\alpha > i_\beta$ dir ve ayrıca $i_1 > 1$ dir. Bunlar P_α^i temel fonksiyonları için düğüm görevi görürler ve şöyle tanımlanırlar:

$$P_\alpha^i = \begin{cases} 0 & 1 \leq i \leq i_{\alpha-2} \\ (i - i_{\alpha-2}) / (i_{\alpha-1} - i_{\alpha-2}) & i_{\alpha-2} \leq i \leq i_{\alpha-1} \\ (i_\alpha - i) / (i_\alpha - i_{\alpha-1}) & i_{\alpha-1} \leq i \leq i_\alpha \\ 0 & i_\alpha \leq i \leq N, \end{cases} \quad (19a)$$

burada $\alpha \geq 2$ ve $i_0 \equiv 1$ dir. $\alpha = 1$ özel durumu için ise P_α^i şu şekilde tanımlanabilir:

$$P_1^i = \begin{cases} (i_1 - i) / i_1 & 1 \leq i \leq i_1 \\ 0 & i_1 \leq i \leq N. \end{cases} \quad (19b)$$

Böylece temel fonksiyon P_α^i $i_{\alpha-1}$ düğümünde tek değere sahiptir. i , $i_{\alpha-2}$ ve i_α komşu düğümlerine yaklaşırsa, P_α^i de lineer olarak sıfıra gider ve bu erim dışında sıfırdır. Şekil-3(a) $\nu = 4$ için temel fonksiyonların mümkün olan bir kümesini gösterir.

(10) denkleminin ayrımını gerçekleştirmede, a_α katsayıları için bir formüle ihtiyacımız vardır. Bu P_α^i nin lineer kombinasyonları olan Q_α^i temel fonksiyonlarının

linmesiyle bulunabilir.

$$Q_{\alpha}^i = \sum_{\beta} B_{\alpha\beta} P_{\beta}^i, \quad (20)$$

$$\sum_i Q_{\alpha}^i P_{\beta}^i = \delta_{\alpha\beta}. \quad (21)$$

$B_{\alpha\beta}$ matrisi, $R_{\alpha\beta}$ gibi bir pozitif simetrik matrisin inversidir. $R_{\alpha\beta}$ matrisi şöyle tanımlanır:

$$R_{\alpha\beta} = \sum_i P_{\alpha}^i P_{\beta}^i \quad (22)$$

11) ve (21) denklemlerinden şu yazılır:

$$\sum_i Q_{\alpha}^i \Delta\gamma_i = 0 \quad (23)$$

Böylece (10) denkleminde de $\Rightarrow$

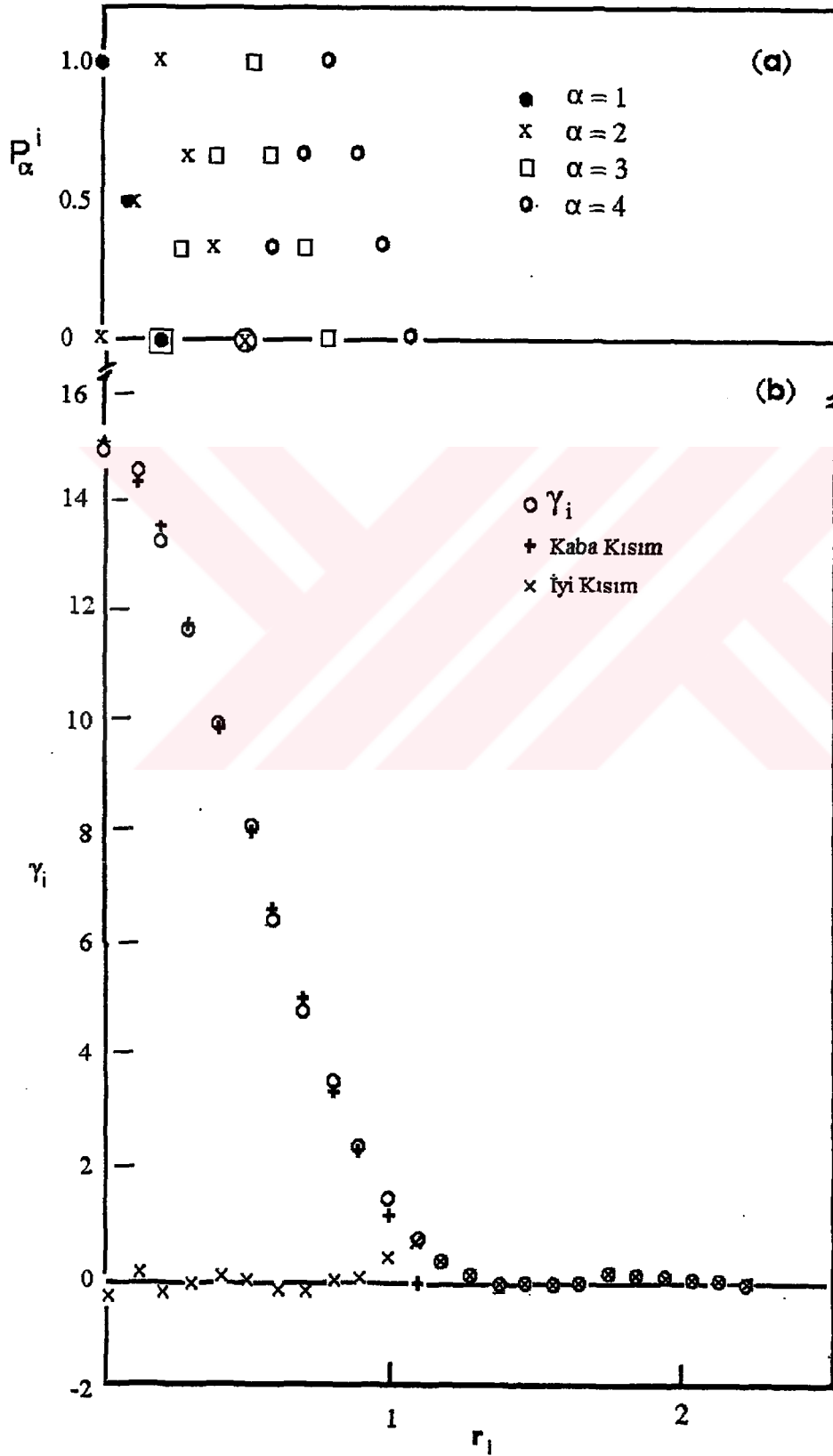
$$a_{\alpha} = \sum_i Q_{\alpha}^i \gamma_i \quad (24)$$

yazılır.

Bu ayırmaya bir örnek olarak, Şekil-3(b) de γ_i nin Şekil-3(a) daki temel fonksiyonlar kullanılarak ayrıştırılmış hali gösterilebilir. Bu şunu gösterir; çok fazla olmayan temel fonksiyonlarla bile iyi kısmın bileşenleri kalan kısımdan daha küçük yapılabilir.

ŞEKİL-3(a) $\nu = 4$ için temel fonksiyonların mümkün olan bir seçimi

ŞEKİL-3(b) 3(a) daki temel fonksiyonlar kullanılarak γ_i istenilen bir ipik yüksek yoğunlukta kaba ve iyi kısımlarına ayrıştırılmıştır.



2.3. Jacobian'ın Hesaplanması

(16) denklemiyle $J_{\alpha\beta}$ jacobian matrisi tanımlanmıştır. d_α nın tanımlanmasından şunu elde ederiz:

$$\begin{aligned} J_{\alpha\beta} &= \delta_{\alpha\beta} - \partial a'_\alpha / \partial a_\beta \\ &= \delta_{\alpha\beta} - \sum_i \partial a'_\alpha / \partial \gamma_i \cdot \partial \gamma_i / \partial a_\beta, \\ &= \delta_{\alpha\beta} - \sum_{ij} Q_\alpha^i \partial \gamma'_i / \partial \gamma_j P_\beta^j \end{aligned} \quad (25)$$

burada (10) ve (24) denklemleri kullanılır. $\partial \gamma'_i / \partial \gamma_j$ için bir ifade (18) denklemin-den elde edilebilir,

$$\frac{\partial \gamma'_i}{\partial \gamma_j} = \sum_{m=0}^{1/2N-1} \frac{\partial \gamma'_i}{\partial \tilde{\gamma}_m} \cdot \frac{\partial \tilde{\gamma}_m}{\partial \tilde{c}_m} \cdot \frac{\partial \tilde{c}_m}{\partial c_j} \cdot \frac{\partial c_j}{\partial \gamma_j} \quad (26)$$

küçük bir işlemle (26) denklemindeki dört çarpan yerine yeni ifadeler yazılarak PY hali için aşağıdakini elde ederiz,

$$\partial \gamma'_i / \partial \gamma_j = (\delta r r_j / \pi r_i) (\exp(-\beta \phi_j) - 1) (D_{i-j} - D_{i+j}) \quad (27)$$

$i \neq 0$ için burada

$$D_l = \delta k \sum_{m=0}^{1/2N-1} \cos k_m r_l \left(\frac{2n\tilde{c}_m}{1-n\tilde{c}_m} + \left(\frac{n\tilde{c}_m}{1-n\tilde{c}_m} \right)^2 \right) \quad (28)$$

= 0 özel hali için şunu elde ederiz:

$$\partial\gamma'_0/\partial\gamma_j = (2\delta rr_j/\pi)(\exp(-\beta\phi_j) - 1)E_j \quad (29)$$

burada

$$E_j = \delta k \sum_{m=0}^{1/2N-1} k_m \sin k_m r_j \left(\frac{2n\tilde{c}_m}{1 - n\tilde{c}_m} + \left(\frac{n\tilde{c}_m}{1 - n\tilde{c}_m} \right)^2 \right) \quad (30)$$

HNC hali için (27) ve (29) denklemlerinde $\exp(-\beta\phi_j) - 1$ yerine $\exp(-\beta\phi_j + \gamma_j)$ konulur. Böylece tüm $\partial\gamma'_i/\partial\gamma_j$ türevleri (28) ve (30) fourier transformasyon denklem çiftinden elde edilebilir. Gerçekte fast fourier transformasyon tekniği, (18) transformasyonuyla bu iki transformasyondan birine izin verir, bu her halde yapılacaktır. $\partial\gamma'_i/\partial\gamma_j$ nin hesabı önemsiz bir orana sahip olduğundan ihmal edilebilir. $J_{\alpha\beta}$ nın hesaplanmasındaki zorluk (27) denklemindeki çift toplamdan kaynaklanır. Hatta sadece i ve j nin küçük bir eriminde Q_α^i ve P_α^i temel fonksiyonlarının sıfırından farklı olmasından dolayı bu çok fazla değildir.

Eğer $\nu = 10$ ise $J_{\alpha\beta}$ nın inversini de içeren hesaplama önemsizdir. Bu uygun bir biçimde yapılabilir.

3. METODUN İNCELENMESİ

Yazdığımız bilgisayar programı için kullandığımız çift potansiyeli, yüklü yumuşak küre potansiyeli olarak adlandırılır ve

$$\phi(r) = \frac{z^2 e^2}{r} + f(\rho) e^{\frac{R-r}{\rho}} \quad (31)$$

formuna sahiptir [Tatlıpınar et al.1992]. Bu potansiyel ile incelenen sistem, elektriksel olarak pozitif yüklü iyonlar arasındaki etkileşmeyi vermektedir. Gözönüne alınan modele göre iyonlar küresel olarak simetrik ve belli bir esnekliğe sahiptir. Bu esneklikten dolayı eğer küreler yeteri kadar bir kuvvet ile yakınlaştırılıp, sıkıştırılırsa, bir miktar iç içe geçebilirler (overlap yapabilirler).

Çift potansiyelde kullanılan parametreler ise deneysel ve kuramsal olarak elde edilen değerlerden alınmıştır. Buna göre, z iyonun yükü, e elektron yükünün değeri, f üst üste gelme (overlap) parametresi, ρ iyonların sertliği ve R de iyonun yarıçapı olmak üzere Y^{3+} (Yitrium) akışkanı için sayısal değerleri şöyledir: $z=+3$, $e=1$, $f=0.05 \text{ e}^2 \text{ Å}^{-2}$, $\rho = 0.007 \text{ Å}^0$, $R = 1.2 \text{ Å}^0$ ve sıcaklık $T=1020 \text{ K}$ dir. Bilgisayar programı geniş sıcaklık aralığında test edilmiştir. Burada amaç, iyi bilinen potansiyeller için sayısal çözümler elde etmek değil, yalnızca yöntemi test etmektir.

Şimdi genel olarak programın işleyişini özetleyelim. Dört temel fonksiyondan daha az kullanılırsa memnun edici olmayan şu sonuca varılır: Çoğunlukla işleyiş Picard yöntemine bırakılır. O zaman yakınsama yavaş veya yoktur. Açıkça, eğer temel fonksiyonların sayısını arttıırırsak, iyileştirme döngülerinin sayısı azalacaktır.

3ununla birlikte her bir adımda jacobian matrisin inversini almak ve hesaplamak için gereken hesaplama zamanına da bir fiat ödenecektir. Bu yüzden temel fonksiyonların sayısını altı ve on arasında almak mantıklı olacaktır. Ama bilindiği gibi temel fonksiyonların seçimi çok önemli değildir.

Rapor edilen tüm sonuçlar $N=513$, $\delta r/R = 0.005$ içindir. Ardışık NR döngüsü

$$\sqrt{(\sum_{\alpha} d_{\alpha}^2)} < 10^{-5} \quad (32)$$

olduğunda birbirine yaklaştığı kabul edilir. Her bir mertebedeki çözümün tam olarak doğruluğunun bir ölçüsü olarak ise,

$$\eta = \sqrt{(\delta r \sum_i (\gamma'_i - \gamma_i)^2)} \quad (33)$$

eşitliğiyle tanımlanan η kullanılır. Burada γ'_i ve γ_i bir elementer döngünün giriş ve çıkışıdır (input-output) Yaklaşımın oranını tanımlamak için, her birindeki NR döngülerinin sayısı ile birlikte her bir iyileştirme döngüsünün sonundaki η nın değeri Tablo-1 de görülmektedir. Elde edilen sonuçlarda $\eta < 10^5$ olduğu zaman yaklaşımın sağlandığı kabul edilmiştir.

Yöntemin kararlı çalışması için, çok hatalı başlangıç yaklaşımından başlamanın etkisini biliyoruz. Ama buna karşı basit bir önlem almadıkça NR yönteminin sapabileceğinden de söz edilmelidir. Bu, her bir döngüdeki a_{α} nın δa_{α} değişiminin limitini içerir. (15) denklemleriyle verilen tüm değişim aşırı olabilir. Aşırı değişimi önlemek için aşağıdaki koşul test edilmelidir.

$$\sum_{\alpha} \delta a_{\alpha}^2 < \Delta^2 \quad (34)$$

Burada Δ seçilen bir sabittir. Bu yeterli olmazsa, aşağıdaki yer değiştirme yapılır.

$$\delta a_\alpha \rightarrow \frac{\Delta}{\sqrt{(\sum_\alpha \delta a_\alpha^2)}} \delta a_\alpha \quad (35)$$

Böylece δa_α nın normu Δ ile sınırlandırılır. Δ nın seçimi hız ve güven arasında bir uyum gerektirir. Bununla birlikte hesaplamaların ilk aşamalarında (33) koşulu aşarlamadığından güven aralığında hata oluşabilir. Döngülerin sayısı başlangıç yaklaşımının durumuna bağlı değildir. Çok zayıf bir yaklaşım verirse, çözüm bölgesinde yaklaşımları bulmak için bir kaç ekstra döngüye ihtiyaç duyulabilir. Ayrıca logru başlangıç yaklaşımlarının oluşturulmasında belirli bir özellik yoktur.

ıcaklık T=1E9 için

ileştirme Döngü No	1	2	3	4
ileştirme Döngüsündeki R Döngü Sayısı	6	4	2	1
ileştirme Döngüsünün sonundaki η	7.1141E-3	4.5755E-6	1.6381E-9	3.7431E-12

ıcaklık T=6E3 için

ileştirme Döngü No	1	2	3	4	5	6	7
ileştirme Döngüsündeki IR Döngü Sayısı	8	6	5	4	2	2	1
ileştirme Döngüsünün sonundaki η	6.53E-3	4.50E-4	8.66E-6	1.13E-7	5.25E-9	1.24E-10	4.52E-13

TABLO-1

Aşağıdaki noktalar önemlidir: (1) Yaklaşım için gerekli döngülerin sayısı termodinamik halin çok güçlü bir fonksiyonu değildir. Yüksek yoğunluklardaki sayının yaklaşık üç katıdır. Geleneksel yöntemde iterasyonların sayısındaki artış çok fazladır. (2) Her bir potansiyel için, temel fonksiyonların seçimi ve başlangıç yaklaşımı oluşturma yöntemi daima aynıdır. (3) Rapor edilen durumların hiçbirisinde sonuç çözüme doğru bir başlangıç yaklaşımı ile gidilmemiştir.



4. SONUÇ

NR ve Picard yöntemlerinin bir kombinasyonu, sıvı yapının integral denklemlerini çözmek için güçlü bir hesaplama planı verir. Bu planın hızlı yakınsadığı, yararlı olduğu ve bilgisayar zamanının oldukça kısa sürdüğü yüklü yumuşak küre potansiyeline uygulanan testlerle gösterildi. Elde edilen yöntemin temel avantajı kararlı oluşudur. Bu da şu anlama gelir, başlangıç yaklaşımlarını oluşturmada herhangi bir zahmete girmek gereksizdir.

Yöntemin en büyük dezavantajı programlamayı gerektirmesidir. Bilgisayar programı, temel fonksiyonları ve onların bileşenlerini oluşturmayı gerektirir. Bu ayrıca γ_i nin iyi ve kaba kısımlarının elle işlenmesini ve her bir döngüde jacobian matrisin hesaplanmasını gerektirir.

Yöntemin uygulamasının en verimli alanı yoğun kulomb sistemleri için (örneğin: erimiş tuzlar) integral denklemlerinin çözümü olduğu görülmektedir. Bu geleneksel yöntemin ağır güçlüklerle karşı karşıya kaldığı durumdur [Hansen et al,1975-Adams,1976-Larsen,1978]. Ayrıca HNC denklemleri son derece iyi doğrulukla yakınsarken, PY için bu söylenemez.

EK.1. NEWTON-RAPHSON YÖNTEMİ

Lineer olmayan denklemlerin çözümü için kullanılan NR yöntemi sayısal analiz kitaplarında verilmiştir [Çağal,1989]. Burada yöntemin ana fikrini özetleyeceğiz.

Yöntem geometrik anlamda çok kolay tanımlanabilir. Lineer olmayan bir $f(x)$ fonksiyonunun x^* kökünü bulmak istiyorsak ve bu fonksiyonun bir $x^{(1)}$ başlangıç değerini kabul edip, bu noktadan hareketle $f(x)$ fonksiyonunun tanjantını çizerek, Şekil-4 de görüldüğü gibi apsisi kestiği yerdeki $x^{(2)}$ noktasını bulabiliriz.

$$f^{(1)} = -\left(\frac{df}{dx}\right)_{x=x^{(1)}}(x^{(2)} - x^{(1)}) \quad (E1)$$

N tane $\{x_j\}$ değişkeninin N tane $f_i(\{x_j\})$ fonksiyonunun bir kümesi olarak bunu genelleştirmek doğru olur. $\{x_j^{(1)}\}$ başlangıç yaklaşımı aşağıdaki ifadeye göre $\{x_j^{(2)}\}$ yaklaşımını verir.

$$f_i^{(1)} = -\sum_j J_{ij}(x_j^{(2)} - x_j^{(1)}), \quad (E2)$$

burada

$$J_{ij} = \frac{\partial f_i}{\partial x_j} \quad (E3)$$

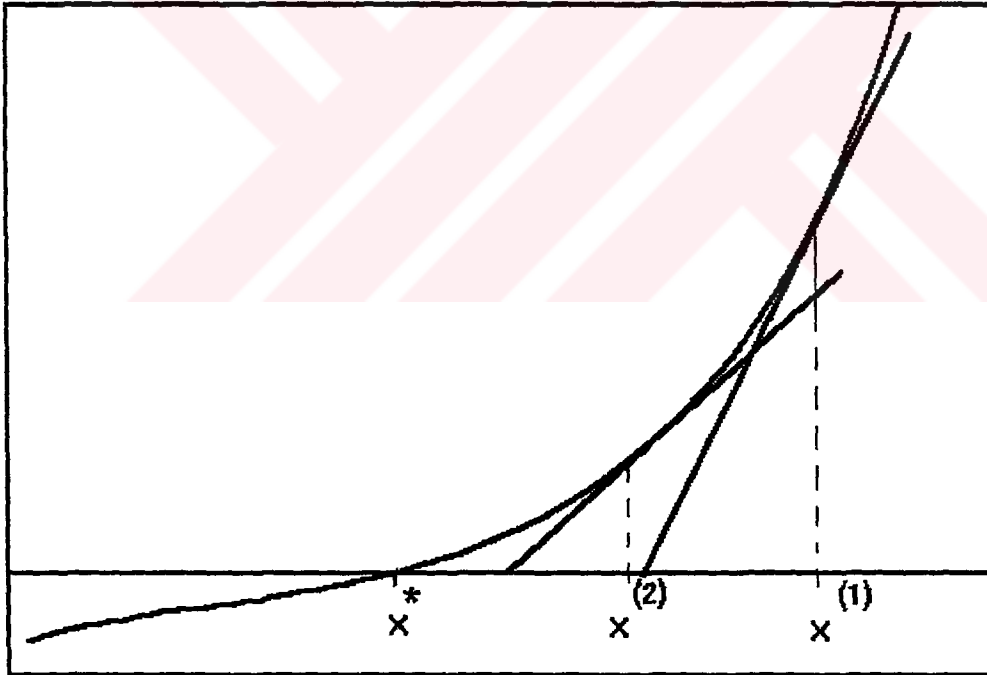
şeklinde ilk değerler hesaplanmıştır. J singüler olmayan bir matristir.

NR yönteminin en önemli özelliklerinden biri, genel koşulların uygunluğu altında bulunan quadratik yaklaşım olmasıdır. Eğer (E2) denklemi itere edilirse, $x_j^{(n)}$ yaklaşımları $\{x_j^*\}$ gerçek çözümüne yeterince yaklaşır.

Bazı α pozitif sabit katsayılar için şu koşul geçerlidir:

$$\sqrt{\left[\sum_j (x_j^{(n+1)} - x_j^*)^2\right]} < \alpha \sum_j (x_j^{(n)} - x_j^*)^2 \quad (E4)$$

Bu, yaklaşımın gerçek çözüme çok hızlı yakınsadığı anlamına gelir.



ŞEKİL-4

EK.2. BİLGİSAYAR PROGRAMI

```
PROGRAM YONTEM
PARAMETER(M=9,NU=20,NU3=3*NU)
PARAMETER(NX=2**M+1)
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
DIMENSION CR(NX),F(NX),CRK(NX),AUSF(NX)
DIMENSION GAMK(NX),GAM(NX),GPR(NX),GDR(NX)
DIMENSION DGAM(NX),AA(NU),DELTA(NX)
DIMENSION AAY(NU),HJAC(NU3,NU3)
DIMENSION D(NU3),DY(NX)
DIMENSION BP(NU,NU),POT(NX)
COMMON/COM1/GDR
COMMON/COM2/GCT,IPP,KOD
COMMON/DEYENI/DY
COMMON/YOGUNL/Z1,Z1SQ,RO1
COMMON/FRFR/AUSF
COMMON/GAMGAM/GAM
COMMON/GPRGPR/GPR
COMMON/CRCRCR/CR
COMMON/COM3/CRK,DELTA
COMMON/ALALAL/S11
COMMON/GKGKGK/GAMK
COMMON/FFFF/F
COMMON/COM4/G,PI,DELK,RO,DLTR,POT
COMMON/NODE/BP,RNOD,NOD
COMMON/NODE1/L1,L2,NHALF
COMMON/HJACHJAC/HJAC
COMMON/POTPAR/FPOT,TROP1,TR1
OPEN(16,FILE='OUT7',STATUS='UNKNOWN')
OPEN(15,FILE='RES7',STATUS='UNKNOWN')
OPEN(3,FILE='INP7',STATUS='UNKNOWN')
OPEN(7,FILE='PTNSYL7',STATUS='UNKNOWN')
REWIND 3
REWIND 7
TOLD=1.D-10
TOLG=1.D-10
DLTR=0.05
Z1=+3.
S11=0.01
PRINT*, 'HNC [1] - PY [0] ?'
READ*,KOD
PRINT*, 'SICAKLIK (K)=? '
READ*,TEMP
DENS=2.56
WETMOT=195 264
```

```

AVOQN=6.022E23
ROMOL=DENS*AVOQN/WEIMOL
RONUM=ROMOL*4
PI=ACOS(-1.D0)
ALENG=(3./(4*PI*RONUM))**(1./3.)
ALENGA=ALENG*1.E8
BOLTZK=1.381771E-16
ECHARG=4.8E-10
G=ECHARG**2/(BOLTZK*TEMP*ALENG)
NJFREQ=25
RO=(3./4./PI)
RO1=1.
RO1=RO*RO1
REWIND 3
REWIND 7
DO 1 I=1,NX
POT(I)=0.D0
GAM(I)=0.D0
1  CONTINUE
DO 993 I=1,NX
READ(3,*,END=991)GAM(I)
993 CONTINUE
991 CONTINUE
DO 93 I=1,NX
READ(7,*,END=51)POT(I)
93 CONTINUE
51 CONTINUE
POT(1)=2000.D6
FPOT=0.05007
RY=1.12
ROY=0.007
FPOT=FPOT*ALENGA**2
ROY=ROY/ALENGA
RY=RY/ALENGA
TR1=2*RY
TROP1=2*ROY
DO 994 I=2,NX
AR=(I-1)*DLTR
IF((TR1-AR)/TROP1.GT.80.) THEN
POT(I)=G*FPOT*TROP1*EXP(80.)
ELSE
POT(I)=G*(FPOT*TROP1*EXP((TR1-AR)/TROP1))
END IF
994 CONTINUE
CALL BPHESES
ITT=0
CALL AHES(GAM,AA)

```

```

CALL DELGAM(GAM,AA,DGAM)
ISAY1=0
ISAY2=0
43  DKONT=0.DO
    DKONT1=0.DO
    DO 158 I=1,L2-1
      GAM(I)=DGAM(I)
    DO 159 J=1,NU
C    PAI NIN HESAPLANMASI
      I1=NOD*(J-2)
      I2=I1+NOD
      I3=I2+NOD
      IF(I.GT.I1.AND.I.LE.I2) THEN
        PAI=(I-I1)/RNOD
      ELSE
        IF(I.GT.I2.AND.I.LT.I3) THEN
          PAI=(I3-I)/RNOD
        ELSE
          PAI=0.DO
        END IF
      END IF
159  GAM(I)=GAM(I)+AA(J)*PAI
158  CONTINUE
    CALL CRHES
    CALL CDK
    CALL GAMNEW
    ITT=ITT+1
    CALL AHES(GPR,AA)
    DO 40 I=1,NU
      D(I)=AA(I)-AAY(I)
      DKONT1=DKONT1+D(I)**2
40  CONTINUE
    DKONT=DKONT1
    PRINT 233,DKONT
233  FORMAT(1X,'NR DON.KONT= ',D11.5)
    ISAY=ISAY1+1
    IF(DKONT.GT.TOLD) THEN
      MM=MOD(ITT+1,NJFREQ)
      IF (ITT.LT.3.OR.MM.EQ.0) CALL JACO
      DO 41 I=1,NU
        DO 42 J=1,NU3
          AA(I)=AA(I)-HJAC(I,J)*D(J)
42  CONTINUE
41  CONTINUE
    GO TO 43
    END IF
    GTOL=0.DO

```

```

CALL DELGAM(GPR,AAY,DY)
DO 2 I=1,NX
GTOL=GTOL+(DY(I)-DGAM(I))**2
DGAM(I)=DY(I)
GAM(I)=GPR(I)
2 CONTINUE
PRINT *, 'NR DONGU SAYISI=', ISAY1
PRINT *, 'ETA =', GTOL
ISAY2=ISAY2+1
IF(GTOL.GT.TOLG) THEN
ISAY1=0
GO TO 43
END IF
PRINT *, 'IYILESTIRME DONGUSU SAYISI=', ISAY2
WRITE(15,*) 'GAMMA =', G
WRITE(15,*) 'RO      =', RO/3.*4.*PI
CALL CRHES
CALL CDK
CALL GAMNEW
WRITE(15,58)
GDR(0)=1.D0
DO 79 I=1,200
R=(I-1)*DLTR
AR=R*ALENGA
WRITE(15,56)R,CR(I),AR
16 FORMAT(1X,2(3X,F9.4),3X,F8.3)
18 FORMAT(9X,'R',11X,'CR',11X,'AR')
19 CONTINUE
WRITE(16,6000)GAM(1),DLTR,G
000 FORMAT(E18.12,1X,2(F8.4))
001 FORMAT(E18.12)
DO 78 I=2,NX
WRITE(16,6001)GAM(I)
8 CONTINUE
COFIN=PI*DLTR
XPR=0.D0
CR(NX)=0.D0
DO 600 I=NX-1,IPP+1,-1
AR=(I-1)*DLTR
XAT=AR*CR(I)+Z1SQ*G
CR(I)=CR(I+1)+COFIN*(XAT+XPR)
XPR=XAT
00 CONTINUE
DO 601 I=IPP,1,-1
AR=(I-1)*DLTR
XAT=AR*(CR(I)+GCT)+Z1SQ*G
CR(I)=CR(I+1)+COFIN*(XAT+XPR)

```

```

XPR=XAT
601  CONTINUE
      DO 605 I=1,IPP
      AR=(I-1)*DLTR
      CR(I)=CR(I)-PI*GCT*(S11**2-AR**2)
605  CONTINUE
      STOP
      END
C***** ALT  PROGRAMLAR *****
      SUBROUTINE BPHE
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      PARAMETER(M=9,NU=20)
      PARAMETER(NX=2**M+1)
      PARAMETER(NPOINT=2**M)
      DIMENSION LWR(NU),BP(NU,NU)
      DIMENSION AUSF(NX),POT(NX)
      COMMON/COM4/G,PI,DELK,RO,DLTR,POT
      COMMON/ALALAL/S11
      COMMON/YOGUNL/Z1,Z1SQ,RO1
      COMMON/FRFR/AUSF
      COMMON/NODE/BP,RNOD,NOD
      COMMON/NODE1/L1,L2,NHALF
      COMMON/COM2/CJM,IPP
      DO 33 I=1,NU
      DO 34 J=1,NU
      BP(I,J)=0.D0
34  CONTINUE
33  CONTINUE
      NHALF=6
      NOD=2*NHALF
      RNOD=NOD
      BP(1,1)=(RNOD+1.)*(2.*RNOD+1.)/6./RNOD-1.
      BP(2,2)=(2.*RNOD**2+1.)/3./RNOD
      BP(1,2)=(RNOD**2-1.)/6./RNOD
      DO 102 I=2,NU
      BP(I,I-1)=BP(1,2)
      BP(I,I)=BP(2,2)
      BP(I-1,I)=BP(1,2)
102 CONTINUE
      CALL MATIN2(BP,NU,NU,MDIM,0,LWR,NERROR,DET)
      Z1SQ=Z1*Z1
      AUS1=G/DLTR
      DL108=DLTR*1.08
      DO 3 I=2,NX
      RAR=(I-1)*DL108
3  AUSF(I)=AUS1/(I-1)
      DO 1 I=2,NX

```

```

      AR=(I-1)*DLTR
      IF (AR.LE.S11+1.D-8) IPP=I
      POT(I)=POT(I)+G*Z1SQ/AR
1     CONTINUE
      L1=(NU-1)*NOD
      L2=L1+NOD
      DELK=2.*PI/NPOINT/DLTR
      RETURN
      END

```

C*****

```

      SUBROUTINE CRHES
      PARAMETER(M=9)
      PARAMETER(NX=2**M+1)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION CR(NX),F(NX)
      DIMENSION GAM(NX),AUSF(NX),POT(NX)
      DIMENSION GAMTRP(5),RPOS(5)
      COMMON/GAMGAM/GAM
      COMMON/YOGUNL/Z1,Z1SQ,RO1
      COMMON/CRCRCR/CR
      COMMON/ALALAL/S11
      COMMON/FFFF/F
      COMMON/FRFR/AUSF
      COMMON/COM4/G,PI,DELK,RO,DLTR,POT
      COMMON/COM2/CJM,IPP,KOD
      DO 1 I=2,NX
      IF (KOD.EQ.1) THEN
      ESP1=-POT(I)+GAM(I)
      IF(I.LE.IPP. OR .ESP1.LE.-150.D0)THEN
      CR(I)=-1.-GAM(I)
      ELSE
      CR(I)=EXP(ESP1)-GAM(I)-1.
      END IF
      ELSE IF (KOD.EQ.0) THEN
      ESP1=-POT(I)-1.
      IF(I.LE.IPP. OR .ESP1.LE.-150.D0)THEN
      CR(I)=0.
      ELSE
      CR(I)=EXP(ESP1)*(1.+GAM(I))
      END IF
      END IF
      ARR=(I-1)*DLTR
      F(I)=(CR(I)+Z1SQ*AUSF(I))*ARR
1     CONTINUE
      IF (KOD.EQ.1) THEN
      CR(1)=-1.-GAM(1)
      ELSE

```

1

```

      IF (KOD.EQ.0) CR(1)=0.
      END IF
      F(1)=0.DO
      ISTART=MAX(1,IPP-2)
      DO 44 I=1,5
      GAMTRP(I)=GAM(ISTART+I)-POT(ISTART+I)
      RPOS(I)=(ISTART+I-1)*DLTR
44    CONTINUE
      VG=DDVDIF(GAMTRP,RPOS,5,S11,3)
      EPP=VG
      IF(EPP.LT.-150.DO)THEN
      CJM=0.DO
      ELSE
      CJM=EXP(EPP)
      END IF
      RETURN
      END
C*****
      SUBROUTINE CDK
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      PARAMETER(M=9)
      PARAMETER(NX=2**M+1)
      PARAMETER(NPOINT=2** (M+1) ,NW=3*NPOINT)
      DIMENSION F(NX),CRK(NX),DY(NX)
      DIMENSION GAMK(NX),DELTA(NX),POT(NX),W(NW)
      COMMON/FFCALIS/W
      COMMON/DEYENI/DY
      COMMON/YOGUNL/Z1,Z1SQ,RO1
      COMMON/ALALAL/S11
      COMMON/GOGO/GAMO
      COMMON/COM3/CRK,DELTA
      COMMON/GKGKGK/GAMK
      COMMON/COM2/CJM,IPP
      COMMON/FFFF/F
      COMMON/COM4/G,PI,DELK,RO,DLTR,POT
      COEF2=4.*PI*G
      COEF1=2.*DLTR*DLTR*NPOINT*SQRT(2.** (M-1))
      CCR=CJM*S11
      DO 11 I=2,IPP
11    F(I)=F(I)+CCR
      CALL FTEK(M,F,1,CRK,1)
      AUS11=DELK/2./PI/PI
      COEF=SQRT(2.** (M-1))*AUS11
      GAMO=0.5*COEF2*Z1SQ
      COFAN=4*PI
      DO 2 I=2,NX-1
      RK=(I-1)*DELK

```

```

DEN=RK**2
ESP=DEN/4.6656
IF(ESP.GT.200.DO) THEN
ATT=0.DO
ELSE
ATT=-COEF2/DEN*EXP(-ESP)
END IF
RK1=RK*S11
FDK=(1.-COS(RK1))/DEN
RMOL=COEF1/(I-1)
CRK(I)=CRK(I)*RMOL-COFAN*CCR*FDK+ATT*Z1SQ
RK=(I-1)*DELK
IF(RK.LT.12.) THEN
AUS=EXP(-RK*RK)/RK
ELSE
AUS=0.DO
END IF
DELTA(I)=(1.-RO1*CRK(I))
GK=RO1*CRK(I)**2
DY(I)=GK/DELTA(I)
GK=DY(I)*RK
GAMO=GAMO+RK*GK
GAMK(I)=(GK-COEF2*AUS*Z1SQ)*COEF
2 CONTINUE
DY(NX)=0.DO
DELTA(1)=0.DO
DELTA(NX)=1.
GAMO=GAMO*AUS11
GAMK(1)=0.DO
GAMK(NX)=0.DO
CRK(NX)=0.DO
CRK(1)=0.DO
RETURN
END
C*****
SUBROUTINE GAMNEW
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
PARAMETER(M=9)
PARAMETER(NX=2**M+1)
PARAMETER(NPOINT=2**(M+1),NW=3*NPOINT)
DIMENSION GAMK(NX),GPR(NX),POT(NX),W(NW)
DIMENSION DY(NX)
COMMON/GPRGPR/GPR
COMMON/GKGKGK/GAMK
COMMON/COM4/G,PI,DELK,RO,DLTR,POT
COMMON/GOGO/GAMO
COMMON/DEYENI/DY

```

```

COMMON/YOGUNL/Z1,Z1SQ,RO1
COMMON/FFCALIS/W
CALL FTEK(M,GAMK,1,GPR,1)
DO 1 I=2,NX-1
AR=(I-1)*DLTR
RAR=AR*0.5
GPR(I)=(GPR(I)+Z1SQ*G)/AR
1 CONTINUE
GPR(1)=GAMO
GPR(NX)=G/(NX-1.)/DLTR*Z1SQ
RETURN
END

C*****
SUBROUTINE JACO1
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
PARAMETER(M=9,NU=20)
PARAMETER(NX=2**M+1)
PARAMETER(NPOINT=2**M,NW=3*NPOINT)
DIMENSION CR(NX),F(NX),CRK(NX)
DIMENSION HJAC1(NU,NU)
DIMENSION GAMK(NX),GAM(NX),GDR(NX)
DIMENSION W(NW),BP(NU,NU)
DIMENSION POT(NX),DELTA(NX),TRV1(400)
DIMENSION DY(NX)
COMMON/GAMGAM/GAM
COMMON/CRCRCR/CR
COMMON/COM1/GDR
COMMON/COM3/CRK,DELTA
COMMON/GKGKGK/GAMK
COMMON/FFFF/F
COMMON/NODE/BP,RNOD,NOD
COMMON/NODE1/L1,L2,NHALF
COMMON/COM4/G,PI,DELK,RO,DLTR,POT
COMMON/DEYENI/DY
COMMON/YOGUNL/Z1,Z1SQ,RO1
COMMON/TP/HJAC1
COMMON/FFCALIS/W
COMMON/HJAC/TRV1
COEF=SQRT(2.**(M-1))*DELK
GDR(1)=-COEF
GAMK(1)=0.D0
DO 9 I=2,NX
RK=(I-1)*DELK
SU1=RO1/DELTA(I)*(2.*CRK(I)+DY(I))
GDR(I)=COEF*SU1
GAMK(I)=GDR(I)*RK
9 CONTINUE

```

```

CALL FCIFT(M,GDR,1,F,1)
CALL FTEK(M,GAMK,1,GDR,1)
AB=DLTR*DLTR/PI
DO 12 I=2,L2-1
ABIM1=AB*(I-1)
GAMK(I)=ABIM1*(CR(I)+GAM(I))
12 CONTINUE
TRV1(1)=0.D0
DO 50 I=0,L2-2
I1=I+1
JQ=I1/NOD+1
JQ1=JQ+1
C PAI NIN HESABLANMASI
IQ1=NOD*(JQ-2)
IQ2=IQ1+NOD
IQ3=IQ2+NOD
IF(I1.GT.IQ1.AND.I1.LE.IQ2) THEN
PAI=(I1-IQ1)/RNOD
ELSE
IF(I1.GT.IQ2.AND.I1.LT.IQ3) THEN
PAI=(IQ3-I1)/RNOD
ELSE
PAI=0.D0
END IF
IF(I.EQ.0) THEN
DO 99 J=2,L2-1
TRV1(J)=GDR(J)*2.*GAMK(J)
99 CONTINUE
ELSE
DLTRI=1./(DLTR*I)
DO 29 J=2,L2-1
IND1=ABS(I1-J)+1
IND2=I+J
TRV1(J)=GAMK(J)*(F(IND1)-F(IND2))*DLTRI
29 CONTINUE
END IF
DO 19 IA=1,NU
IF(I1.LT.L1) THEN
QQ=BP(IA,JQ)*PAI+BP(IA,JQ1)*(1.-PAI)
ELSE
QQ=BP(IA,JQ)*PAI
END IF
QQM=QQ*0.5
ISP=L1+NHALF
HJAC1(IA,NU)=HJAC1(IA,NU)-QQM*TRV1(ISP)
DO 18 IB=2,NU
IBM1=IB-1

```

```

JP=NOD*IBM1
JD=JP-NHALF
HJAC1(IA,IB)=HJAC1(IA,IB)-QQ*(TRV1(JP)+0.5D0*TRV1(JD))
HJAC1(IA,IBM1)=HJAC1(IA,IBM1)-QQM*TRV1(JD)
18  CONTINUE
DO 39 JJJ=1,NHALF-1
P1=JJJ/RNOD
IN1=L2-JJJ
IN2=L1+JJJ
HJAC1(IA,NU)=HJAC1(IA,NU)-QQ*(TRV1(IN2)+(TRV1(IN1)-
$TRV1(IN2))*P1)
DO 40 IB=2,NU
IBM1=IB-1
JBP1=JJJ+NOD*(IB-2)
JBM11=NOD*IBM1-JJJ
HJAC1(IA,IB)=HJAC1(IA,IB)-QQ*(TRV1(JBM11)+(TRV1(JBP1)-
$TRV1(JBM11))*P1)
HJAC1(IA,IBM1)=HJAC1(IA,IBM1)-QQ*(TRV1(JBP1)+(TRV1(JBM11)-
$TRV1(JBP1))*P1)
40  CONTINUE
39  CONTINUE
19  CONTINUE
50  CONTINUE
RETURN
END

```

C*****

```

SUBROUTINE AHES(GAM,AA)
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
PARAMETER(M=9,NU=20)
PARAMETER(NX=2**M+1)
DIMENSION GAM(NX),AA(NU),BP(NU,NU)
COMMON/NODE/BP,RNOD,NOD
COMMON/NODE1/L1,L2,NHALF
DO 1 I=1,NU
AA(I)=0.D0
DO 2 J=1,L1-1
IND=J/NOD+1
IND1=IND+1
C  PAI NIN HESAPLANMASI
I1=NOD*(IND-2)
I2=I1+NOD
I3=I2+NOD
IF(J.GT.I1.AND.J.LE.I2) THEN
PAI=(J-I1)/RNOD
ELSE
IF(J.GT.I2.AND.J.LT.I3) THEN
PAI=(I3-J)/RNOD

```

```

ELSE
PAI=0.D0
END IF
END IF
2  AA(I)=AA(I)+(BP(I,IND)*PAI+BP(I,IND1)*(1.-PAI))*GAM(J)
DO 3 J=L1,L2-1
3  AA(I)=AA(I)+BP(I,NU)*GAM(J)*(NU-J/RNOD)
1  CONTINUE
RETURN
END

```

C*****

```

SUBROUTINE DELGAM(GAM,AA,DGAM)
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
PARAMETER(M=9,NU=20)
PARAMETER(NX=2**M+1)
DIMENSION GAM(NX),AA(NU),DGAM(NX),BP(NU,NU)
COMMON/NODE/BP,RNOD,NOD
COMMON/NODE1/L1,L2,NHALF
DO 1 I=L2,NX
1  DGAM(I)=GAM(I)
DO 2 I=1,L1-1
IND=I/NOD+1
IND1=IND+1
C  PAI NIN HESAPLANMASI
I1=NOD*(IND-2)
I2=I1+NOD
I3=I2+NOD
IF(I.GT.I1.AND.I.LE.I2) THEN
PAI=(I-I1)/RNOD
ELSE
IF(I.GT.I2.AND.I.LT.I3) THEN
PAI=(I3-I)/RNOD
ELSE
PAI=0.D0
END IF
END IF
2  DGAM(I)=GAM(I)-AA(IND)*PAI-AA(IND1)*(1.-PAI)
DO 3 I=L1,L2-1
3  DGAM(I)=GAM(I)-AA(NU)*(NU-I/RNOD)
RETURN
END

```

C*****

```

SUBROUTINE JACO2
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
PARAMETER(M=9,NU=20)
PARAMETER(NX=2**M+1)
PARAMETER(NPOINT=2*(M+1),NW=3*NPOINT)

```

```

DIMENSION CR(NX),F(NX),CRK(NX),HJAC1(NU,NU)
DIMENSION GAMK(NX),GAM(NX),GPR(NX),GDR(NX)
DIMENSION TRV1(400),W(NW),BP(NU,NU)
DIMENSION DELTA(NX),POT(NX),DY(NX)
COMMON/GAMGAM/GAM
COMMON/GPRGPR/GPR
COMMON/CRCRCR/CR
COMMON/COM1/GDR
COMMON/COM3/CRK,DELTA
COMMON/GKGKGK/GAMK
COMMON/FFFF/F
COMMON/NODE/BP,RNOD,NOD
COMMON/NODE1/L1,L2,NHALF
COMMON/COM4/G,PI,DELK,RO,DLTR,POT
COMMON/DEYENI/DY
COMMON/YOGUNL/Z1,Z1SQ,RO1
COMMON/TP/HJAC1
COMMON/FFCALIS/W
COMMON/HJAC/TRV1
COEF=SQRT(2.**(M-1))*DELK
GDR(1)=-COEF
GAMK(1)=0.D0
DO 9 I=2,NX
RK=(I-1)*DELK
CCCC=COEF/DELTA(I)/DELTA(I)
GDR(I)=0
9 GAMK(I)=GDR(I)*RK
CALL FCIFT(M,GDR,1,F,1)
CALL FTEK(M,GAMK,1,GDR,1)
AB=DLTR*DLTR/PI
DO 12 I=2,L2-1
12 GAMK(I)=AB*(I-1)*(CR(I)+GAM(I))
TRV1(1)=0.D0
DO 50 I=0,L2-2
I1=I+1
JN=I1/NOD+1
JN1=JN+1
C PAI NIN HESAPLANMASI
IN1=NOD*(JN-2)
IN2=IN1+NOD
IN3=IN2+NOD
IF(I1.GT.IN1.AND.I1.LE.IN2) THEN
PAI=(I1-IN1)/RNOD
ELSE
IF(I1.GT.IN2.AND.I1.LT.IN3) THEN
PAI=(IN3-I1)/RNOD
ELSE

```

```

PAI=0.DO
END IF
END IF
IF(I.EQ.0) THEN
DO 99 J=2,L2-1
TRV1(J)=GDR(J)*2.*GAMK(J)
99 CONTINUE
ELSE
DO 29 J=2,L2-1
IND1=ABS(I1-J)+1
IND2=I+J
TRV1(J)=GAMK(J)*(F(IND1)-F(IND2))/(DLTR*I)
29 CONTINUE
END IF
DO 19 IA=1,NU
IF(I1.LT.L1) THEN
QQ=BP(IA,JN)*PAI+BP(IA,JN1)*(1.-PAI)
ELSE
QQ=BP(IA,JN)*PAI
END IF
ISP=L1+NHALF
HJAC1(IA,NU)=HJAC1(IA,NU)-QQ*TRV1(ISP)*0.5D0
DO 18 IB=2,NU
IBM1=IB-1
JP=NOD*IBM1
JD=JP-NHALF
HJAC1(IA,IB)=HJAC1(IA,IB)-QQ*(TRV1(JP)+0.5D0*TRV1(JD))
HJAC1(IA,IBM1)=HJAC1(IA,IBM1)-0.5D0*QQ*TRV1(JD)
18 CONTINUE
DO 39 JJJ=1,NHALF-1
P1=JJJ/RNOD
IN1=L2-JJJ
IN2=L1+JJJ
HJAC1(IA,NU)=HJAC1(IA,NU)-QQ*(TRV1(IN2)+(TRV1(IN1)-
$TRV1(IN2))*P1)
DO 40 IB=2,NU
IBM1=IB-1
JBP1=JJJ+NOD*(IB-2)
JBM11=NOD*IBM1-JJJ
HJAC1(IA,IB)=HJAC1(IA,IB)-QQ*(TRV1(JBM11)+(TRV1(JBP1)-
$TRV1(JBM11))*P1)
HJAC1(IA,IBM1)=HJAC1(IA,IBM1)-QQ*(TRV1(JBP1)+(TRV1(JBM11)-
$TRV1(JBP1))*P1)
40 CONTINUE
39 CONTINUE
19 CONTINUE
50 CONTINUE

```

```

RETURN
END
)*****
SUBROUTINE JACO3
  IMPLICIT DOUBLE PRECISION(A-H,O-Z)
  PARAMETER(M=9,NU=20)
  PARAMETER(NX=2**M+1)
  PARAMETER(NPOINT=2**(M+1),NW=3*NPOINT)
  DIMENSION CR(NX),F(NX),CRK(NX),HJAC1(NU,NU)
  DIMENSION GAMK(NX),GAM(NX),GPR(NX),GDR(NX)
  DIMENSION TRV1(400),W(NW),BP(NU,NU)
  DIMENSION DELTA(NX),POT(NX),DY(NX)
  COMMON/GAMGAM/GAM
  COMMON/GPRGPR/GPR
  COMMON/CRCRCR/CR
  COMMON/COM1/GDR
  COMMON/COM3/CRK,DELTA
  COMMON/GKGKGK/GAMK
  COMMON/FFFF/F
  COMMON/NODE/BP,RNOD,NOD
  COMMON/NODE1/L1,L2,NHALF
  COMMON/COM4/G,PI,DELK,RO,DLTR,POT
  COMMON/DEYENI/DY
  COMMON/YOGUNL/Z1,Z1SQ,RO1
  COMMON/TP/HJAC1
  COMMON/FFCALIS/W
  COMMON/HJAC/TRV1
  COEF=SQRT(2.**(M-1))*DELK
  GDR(1)=-COEF
  GAMK(1)=0.D0
  DO 9 I=2,NX
    RK=(I-1)*DELK
    SU1=0.0
    GDR(I)=COEF*SU1
9    GAMK(I)=GDR(I)*RK
    CALL FCIFT(M,GDR,1,F,1)
    CALL FTEK(M,GAMK,1,GDR,1)
    AB=DLTR*DLTR/PI
    DO 12 I=2,L2-1
      GAMK(I)=AB*(I-1)*(CR(I)+GAM(I))
12   CONTINUE
    TRV1(1)=0.D0
    DO 50 I=0,L2-2
      I1=I+1
      JQ=I1/NOD+1
      JQ1=JQ+1
C    PAI NIN HESAPLANMASI

```

```

IQ1=NOD*(JQ-2)
IQ2=IQ1+NOD
IQ3=IQ2+NOD
IF(I1.GT.IQ1.AND.I1.LE.IQ2) THEN
PAI=(I1-IQ1)/RNOD
ELSE
IF(I1.GT.IQ2.AND.I1.LT.IQ3) THEN
PAI=(IQ3-I1)/RNOD
ELSE
PAI=0.DO
END IF
END IF
IF(I.EQ.0) THEN
DO 99 J=2,L2-1
99 TRV1(J)=GDR(J)*2.*GAMK(J)
ELSE
DO 29 J=2,L2-1
IND1=ABS(I1-J)+1
IND2=I+J
29 TRV1(J)=GAMK(J)*(F(IND1)-F(IND2))/(DLTR*I)
END IF
DO 19 IA=1,NU
IF(I1.LT.L1) THEN
QQ=BP(IA,JQ)*PAI+BP(IA,JQ1)*(1.-PAI)
ELSE
QQ=BP(IA,JQ)*PAI
END IF
ISP=L1+NHALF
HJAC1(IA,NU)=HJAC1(IA,NU)-QQ*TRV1(ISP)*0.5DO
DO 18 IB=2,NU
IBM1=IB-1
JP=NOD*IBM1
JD=JP-NHALF
HJAC1(IA,IB)=HJAC1(IA,IB)-QQ*(TRV1(JP)+0.5DO*TRV1(JD))
HJAC1(IA,IBM1)=HJAC1(IA,IBM1)-0.5DO*QQ*TRV1(JD)
18 CONTINUE
DO 39 JJJ=1,NHALF-1
P1=JJJ/RNOD
IN1=L2-JJJ
IN2=L1+JJJ
HJAC1(IA,NU)=HJAC1(IA,NU)-QQ*(TRV1(IN2)+(TRV1(IN1)-
$TRV1(IN2))*P1)
DO 40 IB=2,NU
IBM1=IB-1
JBP1=JJJ+NOD*(IB-2)
JBM31=NOD*IBM1-JJJ
HJAC1(IA,IB)=HJAC1(IA,IB)-QQ*(TRV1(JBM31)+(TRV1(JBP1)-

```

```

$TRV1(JBM31))*P1)
HJAC1(IA,IBM1)=HJAC1(IA,IBM1)-QQ*(TRV1(JBP1)+(TRV1(JBM31)-
$TRV1(JBP1))*P1)
40    CONTINUE
39    CONTINUE
19    CONTINUE
50    CONTINUE
      RETURN
      END

```

C*****

```

      SUBROUTINE JACO
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      PARAMETER(NU=20,NU3=3*NU)
      DIMENSION HJAC(NU3,NU3),HJAC1(NU,NU),LWR(NU3)
      COMMON/TP/HJAC1
      COMMON/HJACHJAC/HJAC
      DO 24 I=1,NU
      DO 25 J=1,NU
      HJAC1(I,J)=0.D0
25    CONTINUE
      HJAC1(I,I)=1.
24    CONTINUE
      CALL JACO1
      DO 1 I=1,NU
      DO 2 J=1,NU
      HJAC(I,J)=HJAC1(I,J)
      HJAC1(I,J)=0.D0
2    CONTINUE
1    CONTINUE
      CALL JACO2
      DO 3 I=1,NU
      DO 4 J=1,NU
      HJAC(I+NU,J)=HJAC1(I,J)
      HJAC1(I,J)=0.D0
4    CONTINUE
3    CONTINUE
      CALL JACO3
      DO 5 I=1,NU
      DO 5 J=1,NU
      HJAC(I+2*NU,J)=HJAC1(I,J)
5    CONTINUE
      CALL MATIN2(HJAC,NU3,NU3,MDIM,0,LWR,NERROR,DET)
      RETURN
      END

```

C*****

```

      SUBROUTINE POLINT(F,ARG,MMMM,Z,SUM)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

```

```

        DIMENSION F(20),ARG(20),COF(20)
        MM=MIN0(MMMM,20)
        M=MM-1
        DO 1780 I=1,MM
        COF(I)=F(I)
1780    CONTINUE
        DO 1800 I=1,M
        DO 1790 J=I,M
        JINDEX=MM-J
        INDEX=JINDEX+I
        COF(INDEX)=(COF(INDEX)-COF(INDEX-1))/(ARG(INDEX)-ARG(JINDEX))
1790    CONTINUE
1800    CONTINUE
        SUM=COF(MM)
        DO 1810 I=1,M
        INDEX=MM-I
        SUM=(Z-ARG(INDEX))*SUM+COF(INDEX)
1810    CONTINUE
        RETURN
        END

```

C*****

```

        SUBROUTINE FDUZ
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        COMMON/D700DT/F,RTTWO,N,N2,N4,M
        COMMON/FFCALIS/W(321)
        IF(M.LT.2) GO TO 90
        N4=2**(M-2)
        N2=N4+N4
        N=N2+N2
        F=1.0/SQRT(FLOAT(N))
        DA=4.0*ATAN(1.0)/FLOAT(N2)
        RTTWO=SQRT(2.0)
        NC=N4-1
        IF(NC.LE.0) RETURN
        DO 10 MC=1,NC
        W(MC)=COS(DA*FLOAT(MC))
10    CONTINUE
        RETURN
90    STOP
        END

```

C*****

```

        SUBROUTINE FCIFT (MM,X,IX,Y,iY)
C    BIR REEL CIFT FONKSIYONUN ANALIZI
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DOUBLE PRECISION X(128), Y(128)
        COMMON/D700DT/F,RTTWO,N,N2,N4,M
        COMMON/FFCALIS/W(321)

```

```

DATA MLOC/- 1/
M=MM+1
IF(M.NE.MLOC) CALL FDUZ
MLOC=M
NI=N4
NO=N4+N2+1
KX=1
NFWA=NI
NLWA=NI+N2
DO 10 I=NFWA,NLWA
W(I)=X(KX)
10 KX=KX+IX
NQ=N2
DO 170 L=1,M
NQ2=NQ/2
NQ2M1=NQ2-1
NO1=NO
NO2=NO+N2
NI1=NI
NI2=NI+NQ
W(NO1)=W(NI1)+W(NI2)
W(NO2)=W(NI1)-W(NI2)
IF(NQ2M1) 50,40,20
20 DO 30 IT=1,NQ2M1
NOR1=NO1+IT
NOR2=NO1+NQ2+IT
NIR1=NI1+IT
NIR2=NI2-IT
W(NOR1)=W(NIR1)+W(NIR2)
W(NOR2)=W(NIR1)-W(NIR2)
30 CONTINUE
40 NOR1=NO1+NQ2
NIR1=NI1+NQ2
W(NOR1)=W(NIR1)+W(NIR1)
50 NO1=NO1+NQ
NO2=NO2-NQ
NI1=NI2+NQ
NI2=NI1+NQ
IF(NO1-NO2) 60,120,160
60 NC=0
NS=N4
KC=0
KS=N4
70 CONTINUE
W(NO1)=W(NI1)+W(NI2)
W(NO2)=W(NI1)-W(NI2)
IF(NQ2M1) 110,100,80

```

```

80    NC=NC+NQ
      NS=NS-NQ
      CC=W(NC)
      SS=W(NS)
      DO 90 IT=1,NQ2M1
      NOR1=NO1+IT
      NOR2=NO2+IT
      NIR1=NI1+IT
      NIR2=NI2-IT
      NOI1=NOR1+NQ2
      NOI2=NOR2+NQ2
      NII1=NIR1+NQ
      NII2=NIR2+NQ
      RE=CC*W(NIR2)-SS*W(NII2)
      AI=SS*W(NIR2)+CC*W(NII2)
      W(NOR1)=W(NIR1)+RE
      W(NOR2)=W(NIR1)-RE
      W(NOI1)=+W(NII1)-AI
      W(NOI2)=-W(NII1)-AI
90    CONTINUE
100   KC=KC+NQ2
      KS=KS-NQ2
      NOR1=NO1+NQ2
      NOR2=NO2+NQ2
      NIR1=NI1+NQ2
      NIR2=NI2+NQ2
      W(NOR1)=2.0*(W(KC)*W(NIR1)-W(KS)*W(NIR2))
      W(NOR2)=2.0*(W(KS)*W(NIR1)+W(KC)*W(NIR2))
110   NO1=NO1+NQ
      NO2=NO2-NQ
      NI1=NI2+NQ
      NI2=NI1+NQ
      IF(NO1-NO2) 70,120,160
120   W(NO1)=W(NI1)
      IF(NQ2M1) 160,150,130
130   DO 140 IT=1,NQ2M1
      NOR1=NO1+IT
      NOI1=NOR1+NQ2
      NIR1=NI+NQ+IT
      NIR2=NI+NQ+NQ-IT
      W(NOR1)=+W(NIR1)
      W(NOI1)=-W(NIR2)
140   CONTINUE
150   NOR1=NO1+NQ2
      NIR1=NI+NQ+NQ2
      W(NOR1)=RTTWO*W(NIR1)
160   NT=NI

```

```

      NI=NO
      NO=NT
      NQ=NQ2
170  CONTINUE
      KY=1
      NFWA=NI
      NLWA=NI+N2
      DO 180 I=NFWA,NLWA
      Y(KY)=W(I)*F
180  KY=KY+IY
      RETURN
      END

```

```

      SUBROUTINE FTEK (MM,X,IX,Y,IY)
      BIR REEL TEK FONKSIYONUN ANALIZI
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION X(128),Y(128)
      COMMON/D700DT/F,RTTWO,N,N2,N4,M
      COMMON/FFCALIS/W(321)
      DATA MLOC/- 1/
      M=MM+1
      IF(M.NE.MLOC) CALL FDUZ
      MLOC=M
      NI=N4
      NO=N4+N2+1
      KX=IX+1
      NFWA=NI+1
      NLWA=NI+N2-1
      DO 10 I=NFWA,NLWA
      W(I)=X(KX)
10  KX=KX+IX
      NQ=N2
      DO 170 L=1,M
      NQ2=NQ/2
      NQ2M1=NQ2-1
      NO1=NO
      NO2=NO+N2
      NI1=NI
      NI2=NI+NQ
      IF(NQ2M1) 50,40,20
20  DO 30 IT=1,NQ2M1
      NOR1=NO1+IT
      NOR2=NO1+NQ2+IT
      NIR1=NI1+IT
      NIR2=NI2-IT
      W(NOR1)=W(NIR1)-W(NIR2)
      W(NOR2)=W(NIR1)+W(NIR2)

```

```

30  CONTINUE
40  NOR1=NO1+NQ2
    NIR1=NI1+NQ2
    W(NOR1)=W(NIR1)+W(NIR1)
50  NO1=NO1+NQ
    NO2=NO2-NQ
    NI1=NI2+NQ
    NI2=NI1+NQ
    IF(NO1-NO2) 60,120,160
60  NC=0
    NS=N4
    KC=0
    KS=N4
70  CONTINUE
    W(NO1)=+W(NI1)+W(NI2)
    W(NO2)=-W(NI1)+W(NI2)
    IF(NQ2M1) 110,100,80
80  NC=NC+NQ
    NS=NS-NQ
    CC=W(NC)
    SS=W(NS)
    DO 90 IT=1,NQ2M1
        NOI1=NO1+IT
        NOI2=NO2+IT
        NII1=NI1+IT
        NII2=NI2-IT
        NOR1=NOI1+NQ2
        NOR2=NOI2+NQ2
        NIR1=NII1+NQ
        NIR2=NII2+NQ
        RE=CC*W(NIR2)-SS*W(NII2)
        AI=SS*W(NIR2)+CC*W(NII2)
        W(NOR1)=W(NIR1)-RE
        W(NOR2)=W(NIR1)+RE
        W(NOI1)=+W(NII1)+AI
        W(NOI2)=-W(NII1)+AI
90  CONTINUE
100  KC=KC+NQ2
    KS=KS-NQ2
    NOR1=NO1+NQ2
    NOR2=NO2+NQ2
    NIR1=NI1+NQ2
    NIR2=NI2+NQ2
    W(NOR1)=2.0*(+W(KC)*W(NIR1)+W(KS)*W(NIR2))
    W(NOR2)=2.0*(-W(KS)*W(NIR1)+W(KC)*W(NIR2))
110  NO1=NO1+NQ
    NO2=NO2-NQ

```

```

      NI1=NI2+NQ
      NI2=NI1+NQ
      IF(NO1-NO2) 70,120,160
120   NIR1=NI+NQ
      W(NO1)=W(NIR1)
      IF(NQ2M1) 160,150,130
130   DO 140 IT=1,NQ2M1
      NOI1=NO1+IT
      NOR1=NOI1+NQ2
      NIR1=NI+NQ+IT
      NIR2=NI+NQ+NQ-IT
      W(NOR1)=W(NIR1)
      W(NOI1)=W(NIR2)
.40   CONTINUE
.50   NOR1=NO1+NQ2
      NIR1=NI+NQ+NQ2
      W(NOR1)=RTTWO*W(NIR1)
.60   NT=NI
      NI=NO
      NO=NT
      NQ=NQ2
70    CONTINUE
      Y(1)=X(1)
      KY=IY+1
      NFWA=NI+1
      NLWA=NI+N2-1
      DO 180 I=NFWA,NLWA
      Y(KY)=W(I)*F
80    KY=KY+IY
      Y(KY)=X(KX)
      RETURN
      END

```

```

      INTEGER FUNCTION LOCATF(ARRAY,LENGTH,OBJECT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION ARRAY(2)
      NABOVE=LENGTH+1
      NBELOW=0
0     IF(NABOVE-NBELOW.LE.1)GO TO 200
      MIDDLE=(NABOVE+NBELOW)/2
      IF(OBJECT-ARRAY(MIDDLE))100,180,140
00    NABOVE=MIDDLE
      GO TO 10
40    NBELOW=MIDDLE
      GO TO 10
30    LOCATF=MIDDLE
      GO TO 300

```

```

200  LOCATF=-NBELOW
300  RETURN
    END
C*****
    DOUBLE PRECISION FUNCTION DDVDIF(F,X,N,Z,MMMM)
C    CERN LIBRARY PROGRAM NO E-105.
C    REVISED DEC.,1976 BY F.JAMES TO CHOOSE POINTS
C    SYMMETRICALLY SO THAT RESULTING INTERPOLATED
C    CURVES WILL BE CONTINUOUS
    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
    DIMENSION F(N),X(N)
    DIMENSION ARG(12),COF(12)
    LOGICAL TWICE
    DATA ZERO,MMA/0.,10/
    TWICE=.FALSE.
    M=MINO(MMMM,N-1,MMA)
    IF (X(1).GT.X(N)) GO TO 1020
    IPOINT=IABS(LOCATF(X,N,Z))
    GO TO 1065
.020  I=1
    IF (Z.GT.X(1)) GO TO 1060
    IF (Z.LT.X(N)) GO TO 1055
    COF1=Z-X(1)
    DO 1050 I=2,N
    COF2=Z-X(I)
    IF (COF1*COF2.LE.ZERO) GO TO 1060
.050  CONTINUE
.055  CONTINUE
    I=N
.060  IPOINT=I-1
.065  MM=M+1
    MB2=MM/2
    MEVEN=MB2*2-1
    IL=IPOINT-MB2+1
    IF (IL.LT.1) IL=1
    IU=IL+MEVEN
    IF (IU.GT.N) IL=N-MEVEN
    DO 1075 I=1,MM
    II=I+IL-1
    ARG(I)=X(II)
.075  COF(I)=F(II)
    IF (MEVEN.EQ.M) GO TO 1085
    IU=IL+MEVEN
    IF (IL.LE.1) GO TO 1079
    IF (IU.GE.N) GO TO 1077
    TWICE=.TRUE.
.077  ARG(MM)=X(IL-1)

```

```

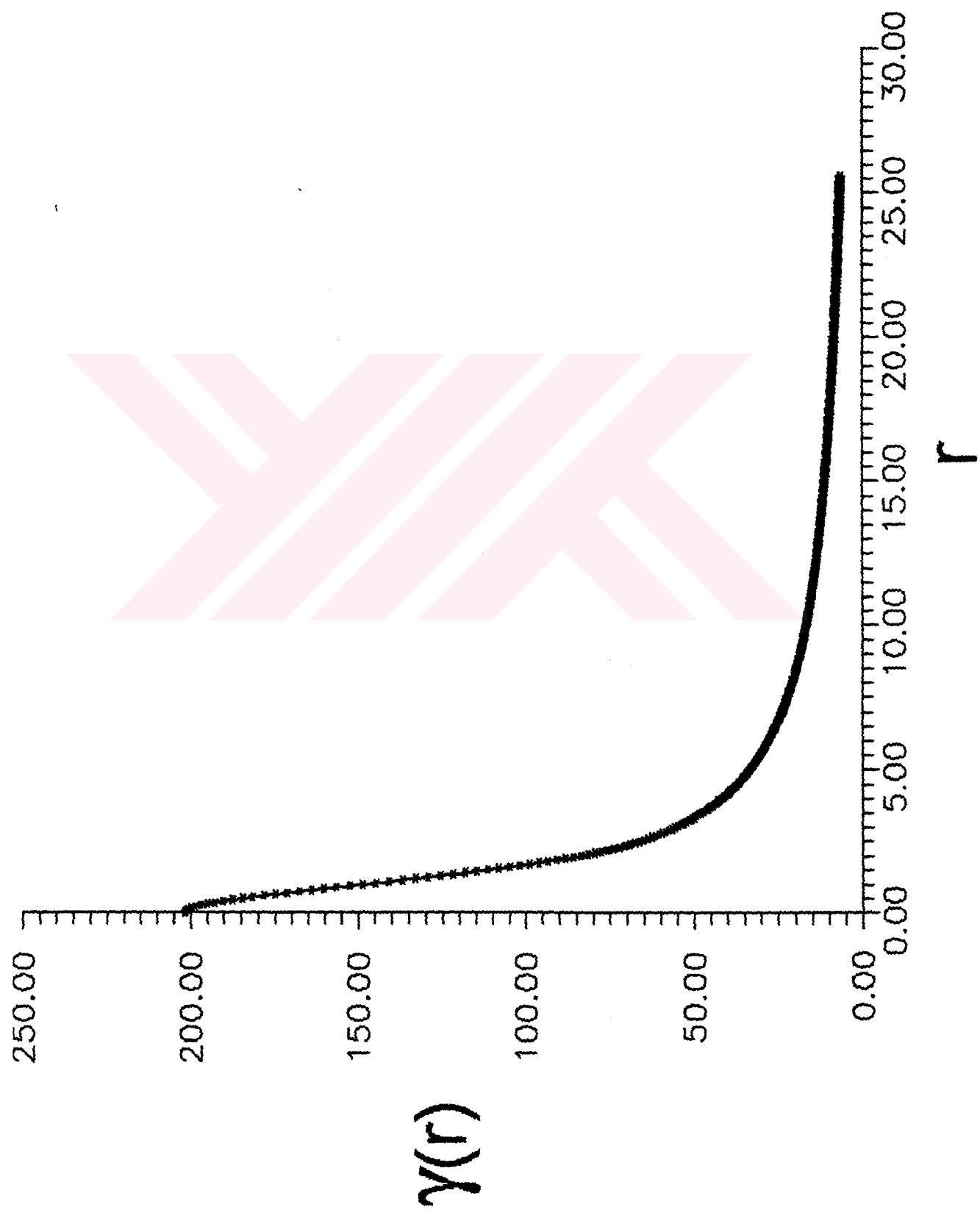
      COF(MM)=F(IL-1)
1079 CONTINUE
1085 CONTINUE
      CALL POLINT(COF,ARG,MM,Z,SUM)
      IF (.NOT.TWICE) GO TO 2000
      ARG(MM)=X(IL-1)
      COF(MM)=F(IL-1)
      CALL POLINT(COF,ARG,MM,Z,SUM2)
      SUM=0.5*(SUM+SUM2)
2000 CONTINUE
      DDVDIF=SUM
      RETURN
      END
;*****
      SUBROUTINE MATIN2(A,IDIM1,N1,IDIM2,N2,INDEX,NERROR,DETERM)
;   MATRISIN TERSINI BULUR
      DOUBLE PRECISION A,DETERM,DETER,PIVOT,SWAP
      DIMENSION A(IDIM1),INDEX(IDIM1)
      DETER=1.0D0
      N=N1
      IEMAT=N+N2
      IDIM=IDIM1
      NMIN1=N-1
      IPIVC=1-IDIM
      DO 11 MAIN=1,N
      PIVOT=0.0D0
      IPIVC=IPIVC+IDIM
      IPIVC1=IPIVC+MAIN-1
      IPIVC2=IPIVC +NMIN1
      DO 2 I1=IPIVC1,IPIVC2
      IF(DABS(A(I1))-DABS(PIVOT)) 2,2,1
      PIVOT=A(I1)
      LPIV=I1
      CONTINUE
      IF(PIVOT) 3,15,3
      ICOL=LPIV-IPIVC+1
      INDEX(MAIN)=ICOL
      IF(ICOL-MAIN) 6,6,4
      DETER=-DETER
      ICOL=ICOL-IDIM
      I3=MAIN-IDIM
      DO 5 I=1,IEMAT
      ICOL=ICOL+IDIM
      I3=I3+IDIM
      SWAP=A(I3)
      A(I3)=A(ICOL)
      A(ICOL)=SWAP

```

```

6      DETER=DETER*PIVOT
      PIVOT=1./PIVOT
      I3=IPIVC+NMIN1
      DO 7 I=IPIVC,I3
7      A(I)=-A(I)*PIVOT
      A(IPIVC1)=PIVOT
      I1=MAIN-IDIM
      ICOL=1-IDIM
      DO 10 I=1,IEMAT
      ICOL=ICOL+IDIM
      I1=I1+IDIM
      IF(I-MAIN) 8,10,8
8      JCOL=ICOL+NMIN1
      SWAP=A(I1)
      I3=IPIVC-1
      DO 9 I2=ICOL,JCOL
      I3=I3+1
9      A(I2)=A(I2)+SWAP*A(I3)
      A(I1)=SWAP*PIVOT
10     CONTINUE
11     CONTINUE
      DO 14 I1=1,N
      MAIN=N+1-I1
      LPIV=INDEX(MAIN)
      IF(LPIV-MAIN) 12,14,12
12     ICOL=(LPIV-1)*IDIM+1
      JCOL=ICOL+NMIN1
      IPIVC=(MAIN-1)*IDIM+1-ICOL
      DO 13 I2=ICOL,JCOL
      I3=I2+IPIVC
      SWAP=A(I2)
      A(I2)=A(I3)
3      A(I3)=SWAP
4      CONTINUE
      DETERM=DETER
      NERROR=0
      RETURN
5      NERROR=MAIN
      DETERM=DETER
      RETURN
      END

```



KAYNAKLAR

- ABERNETHY G.M.,and GILLAN M.J.,1980. Molecular Physics
Vol.39, No.4, 839-847
- ADAMS D.J.,1976. J.Chem.Soc., Faraday Trans. II,72,1372
- ALLEN M.P.,TILDESLEY D.J.,1987. Computer Simulation of Liquids,
first edition, Clarendou Press, Oxford.
- ÇAĞAL B.,1989. Sayısal Analiz. 383-391.
- GILLAN M.J.,1979. Molecular Physics. Vol.38,no.6.1781-1794.
- HANSEN J.P.,and McDONALD I.R.,1975. Phys.Rev. 11,2111.
- HANSEN J.P.,and McDONALD I.R.,1986. Theory of Simple Liquids,
second edition, Academic Press, London.
- LARSEN B.,1978. J.Chem.Phys. 68,4511.
- ORTEGA J.M.,and RHEINBOLDT W.C.,1970. Iterative Solution of
Nonlinear Equations in Several Variables. Chap.7. Academic Press.
- OSTROWSKI A.M.,1966. Solution of Equations and Systems of Equations.
Chap.6, 25. Academic Press.
- TATLIPINAR H.,AKDENİZ Z.,PASTORE G.,and TOSİ M.P.,1992.
J.Phys.: Condens. Matter(4), 8933-8944.
- WATTS R.O.,1973. Statistical Mechanics, Vol.1.,Chap.1. Chemical Society
Specialist Periodical Report)

ÖZGEÇMİŞ

1969 yılında Antalya'da doğdum. İlkokulu Küçükçekmece Yeşilyuva İlkokulu'nda, ortaokul ve liseyi de Küçükçekmece Lisesi'nde bitirdim. Üniversiteyi de Yıldız Üniversitesi Fen-Edebiyat Fakültesi Matematik Bölümü'nde tamamladım. 1991'de yine Yıldız Üniversitesi Fen-Edebiyat Fakültesi Matematik Bölümü'nde uzman olarak çalışmaya başladım.