YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GEMİ ETRAFINDAKİ AKIMIN PANEL METODU KULLANILARAK İNCELENMESİ

Gemi inşaatı ve makinaları mühendisi Levent KAYDIHAN

FBE Makina Fakultesi Anabilim Dalı Gemi inşaatı Mühendisliği programında Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof. Dr. Mesut GÜNER

Prof. Dr. M. Güner
Doç. Dr. H. Yılmaz
Doç. Dr. S. Bal

İSTANBUL 2002

İÇİNDEKİLER

SİMGE LİSTESİ

$\Phi\,(x, y, z, t)$	Hız potansiyeli
$G(x,y,z;\xi,\eta,\zeta)$	Green Fonksiyonu
g	Yer çekimi ivmesi
V_a	Gemi hızı
$\sigma(x,y,z)$	Kaynak şiddeti
$n(x,y,x)$	Panel yüzeyi normalleri
S	Gemi yüzey alanı
R	Gemi toplam direnci
C_p	Basınç katsayısı
C_w	Dalga direnci katsayısı
L	Geminin su hattı boyu
B	Gemi kalıp genişliği
d	Gemi su çekimi

ŞEKİL LİSTESİ

ÖNSÖZ

Bu tez, gemilerin kesin çözümü, ancak deneysel yollarla bulunabilecek olan akım karakteristikleri ve direnç değerlerinin, ön dizayn aşamasında yaklaşık olarak hesaplanabilmesi için kullanılan matematiksel bir metodun incelenmesini amaçlamıştır.

Bu tezin oluşma aşamasında bana her türlü destekte bulunan hocam Prof. Dr. Mesut Güner, sayın Yük. Müh. Yaşar Gül, Yük. Müh. Bülent Şener, Dr. Barbaros M. Okan ve DELTA Denizcilik & Mühendislik çalışanlarına teşekkürü bir borç bilirim.

ÖZET

Bu çalışmada üç boyutlu cisimler etrafındaki potansiyel akım karakteristiklerinin bulunabilmesi için yüzey panel metodu kullanan bir yöntem incelenmiştir. Üç boyutlu cisimler veya bir gemi etrafındaki potansiyel akım Green teoreminden yararlanılarak yüzey üzerine dağıtılmış kelvin kaynakları ile gösterilmekte ve bu kaynakların şiddetleri gemi yüzeyi sınır koşullarını sağlayacak şekilde belirlenmektedir. Gemi yüzeyi üzerindeki kelvin kaynak dağılımı belirlendikten sonra bu kaynak dağılımı kullanılarak gemi yüzeyi etrafındaki potansiyel akım, hızlar, basınç katsayıları ve dolayısıyla gemini direnci hesaplanabilmektedir.

Bu yöntem esas olarak iki kısıma ayrılabilir. Birinci kısımda, su yüzeyinde hareket eden cisimler için formüller çıkarılmış ve çözümler elde edilmiştir. Burada sınır koşulları gemi üzerinde tam olarak, serbest su yüzeyinde ise lineerleştirilmiş olarak kabul edilmiştir (Neumann-Kelvin problemi). İkinci kısımda ise sonsuz akışkan içerisindeki cisimler veya su yüzeyinde çok düşük Froude sayısında hareket eden cisimler için formüller çıkarılmış ve çözümler elde edilmiştir (Double Model problemi).

Her iki kısım içinde genel çözüm yöntemleri verilmiştir ve yöntemler birleştirilerek eksenel simetrik cisimler ve gemiler için nümerik uygulamalar yapılmış ve sonuçları gösterilmiştir.

Yapılan nümerik uygulamalar sonucunda potansiyel akım teorisi kullanılarak yapılan hesaplamaların gemilerin ön dizaynı aşamasında sualtı formunun optimizasyonu için bir yaklaşım metodu olarak kullanılabileceği gösterilmiştir.

Anahtar kelimeler: Panel Metodu, Potensiyel akım, Gemi direnci, Serbest yüzey.

ABSTRACT

In this study, a method for potential flow problem around three dimensional bodies are investigated by using the surface panel method. The potential flow around the three dimensional bodies and ships are formulated by Kelvin sources distributed over the surface by using the Green's functions. These sources are determined to satisfy the ship surface boundary conditions. After the Kelvin sources calculated, velocities, pressure coefficients and therefore the ship resistance values can be obtained.

This method can be investigated in two parts. In first part, the problem for bodies moving on the water surface is formulated and solved by using the linearised free surface and the exact body surface boundary conditions (Neumann-Kelvin problem). In the second part, the problem is investigated for bodies in an infinite fluid or for bodies advancing on the water surface with very low Froude number (Double Model problem).

Solutions methods for both parts are presented and the results of numerical applications for axially symmetric bodies and for ship forms are given.

It is seen in the results of numerical applications, such calculations by using the potential theory can be used as approximation method to optimize the ship's underwater form in the priliminary design stage.

Keywords: Panel Method, Potential flow, Ship resistance, Free surface.

1. GİRİŞ

Serbest su yüzeyi üzerinde hareket eden cisim veya gemilerin etrafındaki akımın ve de oluşturduğu dalga profilinin hesaplanması, araştırmacıların ilgisini uzun bir süredir çekmektedir.

Analitik olarak ele alındığında, serbest su yüzeyi üzerinde hareket eden cisim veya gemilerin etrafındaki akım ancak yapılan belirli kabuller ve yaklaşımlar sonucunda hesaplanabilir. Bu amaç için yapılan en önemli kabuller viskoz olmayan, sıkıştırılamaz ve çevrimsiz akış kabulleridir. Klasik hidrodinamik kurallarından bilindiği gibi, bu durumlar altında hız potansiyeli mevcuttur ve akış alanı içersindeki her noktada bir hız ve bir basınç değerine neden olmaktadır. Ama serbest su yüzeyinin bulunması ve uygulanan analitik metotların limitleri nedeni ile daha fazla kabul ve yaklaşımlar gerekmektedir. Yapılan klasik kabuller altında, Laplace denklemini sağlayan ve harmonic bir fonksiyon olan $\Phi(x, y, z, t)$ hız potansiyelinin çözümü aranmaktadır.

Bir geminin direnci geminin denizde belirli bir hızla ilerlerken karşılaştığı tepkidir. Ancak gemi direncini oluşturan etkenler oldukça karmaşık ve anlaşılması güçtür. İlerlemekte olan gemiden yayılan dalgalar yerçekiminden kaynaklanan bir basınç alanının tepkinin oluşmasında önemli bir payı olduğunu gösterir. Ancak tepkinin tümü yerçekiminden kaynaklanmaz. Deniz suyu viskoz bir akışkan olduğu için gemi ile deniz suyu arasındaki sürtünmeden dolayı ortaya çıkan teğetsel gerilmeler de gemi direncine önemli katkıda bulunurlar. Deniz suyunun viskoz karakteri ayrıca gemi etrafında giderek büyüyen bir sınır tabakanın oluşmasına da neden olur. Ortaya çıkan bu sınır tabakada teğetsel gerilmelerin yanısıra oluşan normal gerilmelerin de dirence katkısı önemlidir.

Gemi direncinin viskoz direnç ve dalga direnci olmak üzere iki temel bileşene ayrılabileceğini ve bu ayrımın deney sonuçlarını değerlendirirken model boyutlarından gemi boyutlarına geçişteki önemini ilk kavrayan araştırmacı Froude olmuştur. Froude tarafından önerilen bu sınıflama ve yöntem günümüze dek gemi direnci ile ilgili tüm deneysel çalışmaların temelini oluşturmaktadır. Ne var ki, Froude tarafından önerilen yöntem gemilerin direncinin analitik olarak hesaplaması konusunda bir katkı sağlamamış, deneysel bir yöntem olarak sınırlı kalmıştır. Bu sınırlamanın aşılması doğrultusundaki ilk adım, yaklaşık on yıl sonra, Michell tarafından sakin deniz koşullarında sabit hız ile ilerleyen bir geminin dalga direncinin hesabı için analitik bir yöntemin geliştirilmesi ile atılmıştır.

Michell problemin çözümü için gemi etrafındaki suyun viskozitesiz olmasının yanı sıra bazı varsayımlara dayanan basitleştirmeler yapmıştır. İlk olarak serbest su yüzeyinde oluşacak dalgaların küçük olacağını varsayarak serbest su yüzeyi koşulunu bozulmamış serbest su yüzeyi civarında lineerleştirmiştir. Ayrıca geminin ince olduğunu varsayarak gemi yüzey koşulunu da geminin orta simetri düzlemi üzerine indirgedikten sonra kısmi türevli diferansiyel denklemin çözümünü çarpanlarına ayırma yöntemi ile elde etmiştir. Çok yakın zamana kadar gemilerin dalga direnci hesabının temelini oluşturan bu varsayımların açıklamasını ve sınır koşullarının lineerleştirilmesini Sabuncu (1962,2000) ayrıntılı olarak vermektedir.

Daha sonraları, genel varsayımlar ve problemin tanımlanması açısından temel farklılıklar getirmemekle birlikte, Havelock (1966) dalga direnci probleminin çözümünde Kelvin kaynak dağılımı kullanarak gemi direncinin analitik olarak hesaplanması konusunda Michellden sonra en önemli katkıyı yapmıştır. Gemi direncinin hesabında Kelvin kaynak dağılımının kullanılması lineer olmayan sınır koşullarının göz önüne alınabilmesini kolaylaştırmış ve ardından daha yüksek mertebeden sınır koşullarının da göz önüne alan çeşitli çalışmalar yapılmıştır. Aldoğan (1977) lineer olmayan dalga direnci teorisini incelerken, konunun teorik temeli ve bu konudaki uygulamaları bir araya getiren, kapsamlı bir çalışma yapmıştır.

Dalga direnci probleminin çözümünde Kelvin kaynak dağılımı kullanılmasının diğer bir avantajı da gemi sınır koşulunun orta simetri düzlemi üzerine indirgenmesi gereğini ortadan kaldırmasıdır. Bilgisayarların gelişmesi ile birlikte herhangi bir cisim etrafındaki potansiyel akımın Hess ve Smith (1967) tarafından cisim üzerine dağıtılmış Rankine kaynakları yardımı ile belirlenmesini takiben Okan (2000), Şaylan (1980) ve diğer bazı araştırmacılar Kelvin kaynaklarını benzer şekilde kullanarak gemi direnci probleminin çözümü için sayısal bir yöntem geliştirmişlerdir. Bu yöntem genel olarak gemi direncinin hesabında eski yöntemlere kıyasla çok büyük bir sayısal hassasiyet getirmemekle birlikte gemi üzerindeki akımın yerel ayrıntılarını belirlemek açısından büyük bir üstünlük sağlamaktadır. Örneğin geminin baş tarafındaki akımı göz önüne alalım. Bu bölgede geminin dip taraflarına doğru akım çoğunlukla aşağı yönlenmektedir. Eğer bu aşağı yönlü akım yeteri kadar büyük ise sintine dönümüne yaklaştıkça yerel olarak çok büyük hızlara ulaşıp sintine dönümünden başlayarak çevrili akımın oluşmasına ve direnç artışına neden olur. Michell'in ince gemi varsayımlarını kullandığımız takdirde gemi sınır koşulunu orta simetri düzlemine indirgediğimiz için bu tür ayrıntıları kaybetmiş oluruz. Bu tür yerel ayrıntıların özellikle sınır tabakanın oluşumunda etkili olduğu göz önüne alınırsa bu yöntemin gemi formundaki bozuklukları düzeltmek konusundaki üstünlüğü tartışılmaz

Birinci dereceden lineerize edilmiş teori ince gemi formu kavramı göz önüne alınarak ortaya çıkmıştır. Son zamanlarda, üç boyutlu kabuk formların yüzeyde gerçek sınır koşullarını sağlayan metotlar daha fazla ilgi çekmeye başlamıştır. Hess ve Smith (1967) serbest su yüzey etkilerini göz ardı ederek problemi çözmüş ve bu konuda başı çekmiştir (Sonsuz akışkandaki double model). Cisim yüzeyi panellere ayrıldıktan sonra, paneller üzerine dağıtılmış kaynak şiddetleri Fredholm integral denklemini çözerek elde edilebilir. Bu yöntem ayrıca lineerize edilmiş serbest su yüzeyi koşulunu da içeren Neumann-Kelvin problemi için de geçerlidir.

Bu çalışmada gemi direncinin gemi yüzeyine dağıtılmış Kelvin kaynaklarından yararlanarak hesaplanması anlatılmaktadır. Bu amaçla C++ programlama dili kullanılarak bir bilgisayar programı geliştirilmiş ve bu program wigley kabuk formuna ve de 5850 Dwt ve 6750 Dwt kapasiteli kimyasal tankerin formlarına uygulanmıştır ve Fluent CFD programı ile karşılaştırılmıştır.. İkinci bölümde göz önüne alınan temel denklemler, üçüncü bölümde çözüm yöntemi ve dördüncü bölümde dalga direnci hesabı anlatılmaktadır. Daha sonraki beşinci bölümde ise yapılan nümerik analizlerin sonuçları verilmiştir.

2. TEORİ

2.1 Serbest Su Yüzeyinde Hareket Eden Cisimler

Sonsuz akışkan içerisinde ve bozulmamış serbest yüzeyde sabit bir hızla hareket eden bir cisim için yapılan kabuller aşağıda verildiği gibidir;

- Akışkan ideal, sıkıştırılamaz ve akış çevrimsizdir,
- Gemi yüzeyi ve paneller düzgün bir yüzeye sahiptir,
- Gemi sabit hızla giderken trim ve meyil oluşmaz.

Gemiye bağlı bir (Oxyz) koordinat sistemi seçilmiştir. Bu sistemde Oxy düzlemi bozulmamış serbest su yüzeyi ile çakışmakta olup, pozitif Ox yönü geminin ilerleme yönünün tersi doğrultusunda, pozitif Oy yönü de sancak tarafından iskele tarafına doğru seçilmiştir. Oxz düzlemi geminin orta simetri düzlemi ile çakışmakta olup Oz ekseni koordinat sisteminin sağ koordinat sistemi olması için serbest su yüzeyinden aşağıya doğru yönlendirilmiştir. Bu koordinat sisteminde $S(x,y,z) = 0$ gemi yüzeyinin denklemi ve $F(x,y,z) = z - \zeta(x,y)$ de serbest su yüzeyinin denklemi olmak üzere Φ hız potansiyeli aşağıdaki şekilde verilmektedir.

$$\Phi(x,y,z) = \Phi_1(x,y,z) + \Phi(x,y,z) \tag{2.1}$$

burada $\Phi_1 = -U.x$, hız potansiyeli ve Φ gemini varlığından dolayı oluşan perturbasyon potansiyelidir.

Gemi yüzeyi dışındaki hacim için süreklilik denklemi Laplace denklemi haline gelir.

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = 0 \qquad -\infty < x,y < \infty \qquad z > 0 \tag{2.2}$$

Geminin sabit ve akımın gemiye doğru geldiği kabulünü yaparak, gemi yüzeyi sınır koşulu elde edilir. Burada, eğer bir akışkan parçacığı gemi yüzeyi üzerinde ise, bütün zaman boyunca orada kalır kabulü yapılmıştır.

$$\frac{\partial\phi}{\partial x} n_x + \frac{\partial\phi}{\partial y} n_y + \frac{\partial\phi}{\partial z} n_z = -V_o n_x \qquad S(x,y,z) = 0 \tag{2.3}$$

Yapılan serbest yüzey ve sonsuz akışkan kabulleri de aşağıda verildiği gibidir.

$$\frac{\partial^2\phi}{\partial x^2} + \frac{g}{V_o^2}\frac{\partial\phi}{\partial z} = 0 \qquad -\infty < x,y < \infty \qquad z = 0 \tag{2.4}$$

$$\frac{\partial \phi}{\partial z} = 0 \qquad -\infty < x, y < \infty \qquad z \Rightarrow \infty \tag{2.5}$$

$$x^2 + y^2 \to \infty \qquad \phi \Rightarrow o(1) \quad x < 0 \qquad \phi \Rightarrow O\left(\left[x^2 + y^2\right]^{-\frac{1}{2}}\right) \quad x > 0 \tag{2.6}$$

Burada (n_x, n_y, n_z) gemi yüzeyinin dış normalinin bileşenleri, V_o da geminin ileri hızıdır.

Green teoremi gereğince ϕ potansiyeli, (ξ,η,ζ) noktasındaki birim şiddetteki bir kaynağın (x,y,z) noktasında yarattığı etkiyi belirten $G(x,y,z;\xi,\eta,\zeta)$ Green fonksiyonu ve $S(\xi,\eta,\zeta) = 0$ gemi yüzeyi üzerine dağıtılmış $\sigma(\xi,\eta,\zeta)$ kaynağı cinsinden

$$\phi(x,y,z) = \frac{1}{4\pi}\iint_S \sigma(\xi,\eta,\zeta) G(x,y,z;\xi,\eta,\zeta) dS(\xi,\eta,\zeta) \tag{2.7}$$

şeklinde yazılabilir. Burada (x, y, z) bölgedeki herhangi bir noktayı, (ξ,η,ζ) de gemi yüzeyindeki bir kaynağın konumunu, $\sigma(\xi,\eta,\zeta)$ kaynak şiddetini ve $dS(\xi,\eta,\zeta)$ de alan elemanını göstermektedir. Green fonksiyonu akışkan bölgesinde geçerli temel denklemin tekil bir çözümü olup ayrıca radyasyon koşulları ile deniz dibinde ve serbest su yüzeyinde sınır koşullarını sağlamalıdır. Böyle bir Green fonksiyonu Havelock tarafından

$$\begin{aligned} G(x,y,z;\xi,\eta,\zeta) = & \frac{1}{\sqrt{a^2+b^2+c^2}} - \frac{1}{\sqrt{a^2+b^2+\bar{c}^2}} - \\ & -\frac{4\kappa_o}{\pi}\int_0^{\frac{\pi}{2}} Sec^2\theta d\theta \int_o^{\infty} \frac{\kappa e^{-\kappa\bar{c}}}{\kappa - \kappa_o Sec^2\theta} Cos(\kappa a Cos\theta) Cos(\kappa b Sin\theta) d\kappa + \\ & + 4\kappa_o \int_0^{\frac{\pi}{2}} e^{-\kappa\bar{c} Sec^2\theta} Sin[\kappa_o a Sec\theta] Cos[\kappa_o b Sin\theta Sec^2\theta] Sec^2\theta d\theta \end{aligned} \tag{2.8}$$

şeklinde verilmiştir. Bu denklemdeki $\kappa_o = g / V_o^2$ karakteristik dalga sayısı, a, b, c de kaynak ile alan noktası arasındaki yer vektörünün bileşenleri $\bar{c}$ de kaynağın imajı olup

$$a = x - \xi \qquad b = y - \eta \qquad c = z - \zeta \qquad \bar{c} = z + \zeta$$

şeklinde tanımlanmıştır. Green fonksiyonunun ilk terimi Rankine kaynağı ikinci terimi de Rankine kaynağının imajını temsil etmektedir. Son iki terim ise serbest yüzey dalgalarını yaratan terimlerdir ve bunlardan tek katlı entegral Green fonksiyonunun radyasyon koşullarını sağlamasını garanti eder. Green fonksiyonunun son iki terimindeki entegrasyon değişkenleri θ ve κ sırası ile geminin ilerleme yönü ile dalganın ilerleme yönü arasındaki açı ile dalga sayısına karşı gelmektedir.

Çok yüksek hızlı gemiler göz önüne alındığında Froude sayısı sonsuza gideceğinden karakteristik dalga sayısı κ_o sıfıra gider ve Green fonksiyonundaki son iki terim düşeceği için potansiyelin yüksek hız limiti uygun olarak elde edilir. Hızın sıfıra gittiği düşük hız limitinde ise κ_o sonsuza gideceği için Green fonksiyonunun bu şekli sorun yaratır. Bu sorunu çözmek için yine Havelock iki katlı intagralin değişik bir şekilde düzenlenerek

$$
\begin{aligned}
G(x,y,z;\xi,\eta,\zeta) = & \frac{1}{\sqrt{a^2+b^2+c^2}} + \frac{1}{\sqrt{a^2+b^2+\bar{c}^2}} - \\
& -\frac{4}{\pi}\int_0^{\frac{\pi}{2}} d\theta \int_0^{\infty} \frac{\kappa e^{-\kappa\bar{c}}}{\kappa-\kappa_o \mathrm{Sec}^2\theta} \mathrm{Cos}(\kappa a \mathrm{Cos}\theta)\mathrm{Cos}(\kappa b \mathrm{Sin}\theta) d\kappa + \\
& + 4\kappa_o \int_0^{\frac{\pi}{2}} e^{-\kappa\bar{c}\mathrm{Sec}^2\theta} \mathrm{Sin}[\kappa_o a \mathrm{Sec}\theta] \mathrm{Cos}[\kappa_o b \mathrm{Sin}\theta \mathrm{Sec}^2\theta] \mathrm{Sec}^2\theta d\theta
\end{aligned} \tag{2.9}
$$

şeklinde yazılabileceğini göstermiştir. Burada iki katlı integral dalga sayısı paydada olduğu için, tek katlı entegral de entegrantın üstel olarak sıfıra gitmesi nedeni ile sıfır olur ve geri kalan terimler de Green fonksiyonunun sıfır hız limitine karşı gelir.

Böylece problem, yukarıdaki gibi tanımlanmış olan $\phi(x,y,z)$ potansiyelindeki $\sigma(\xi,\eta,\zeta)$ kaynak şiddetinin gemi yüzeyi sınır koşulunu sağlayacak şekilde belirlenmesine indirgenmiş olur. Bu amaçla $\phi(x,y,z)$ potansiyelinin tanımını gemi yüzeyi sınır koşulunda yerine koyar ve (ξ,η,ζ) noktası (x,y,z) noktasına yaklaşırken ortaya çıkan tekillik de göz önüne alınırsa $\sigma(x,y,z)$ kaynak şiddeti için aşağıdaki integral denklemini elde ederiz.

$$
\frac{1}{4\pi}\iint_S \sigma(\xi,\eta,\zeta) G_n(x,y,z;\xi,\eta,\zeta) dS(\xi,\eta,\zeta) = -V_o n_x(x,y,z) \tag{2.10}
$$

Bu denklemde $G_n(x,y,z;\xi,\eta,\zeta)$ Green fonksiyonunun (x,y,z) noktasındaki yüzey dış normali doğrultusundaki türevi olup;

$$
\begin{aligned}
G_n(x,y,z;\xi,\eta,\zeta) = & n_x(x,y,z) G_x(x,y,z;\xi,\eta,\zeta) + n_y(x,y,z) G_y(x,y,z;\xi,\eta,\zeta) + \\
& + n_z(x,y,z) G_z(x,y,z;\xi,\eta,\zeta)
\end{aligned} \tag{2.11}
$$

olarak tanımlanmıştır. Burada kullanılmakta olan $G_x(x,y,z;\xi,\eta,\zeta)$, $G_y(x,y,z;\xi,\eta,\zeta)$ ve $G_z(x,y,z;\xi,\eta,\zeta)$ Green fonksiyonunun sırası ile x, y ve z doğrultularındaki kısmi türevlerini göstermektedir. Potansiyelin tanımını ve potansiyel ile hız arasındaki ilişkiyi kullanırsak akışkan bölgesinde hız bileşenleri ;

$$u(x,y,z) = V_o + \iint_S \sigma(\xi,\eta,\zeta) G_x(x,y,z;\xi,\eta,\zeta) dS(\xi,\eta,\zeta)$$

$$v(x,y,z) = \iint_S \sigma(\xi,\eta,\zeta) G_y(x,y,z;\xi,\eta,\zeta) dS(\xi,\eta,\zeta) \qquad (2.12)$$

$$w(x,y,z) = \iint_S \sigma(\xi,\eta,\zeta) G_z(x,y,z;\xi,\eta,\zeta) dS(\xi,\eta,\zeta)$$

şeklinde elde edilir. Bu şekilde hızlar alanı hesaplandıktan sonra basınç dağılımı ve serbest su yüzeyi yükseklikleri lineerleştirilmiş Bernoulli denkleminden;

$$p(x,y,z) = V_o u(x,y,z) \qquad \zeta(x,y) = \frac{V_o u(x,y,0)}{g} \qquad (2.13)$$

şeklinde elde edilirler.

2.2 Sonsuz Akışkan İçerisindeki Cisimler

Bu akış problemini çözebilmek için yapılabilecek en basit yaklaşım, cisim hareketinin serbest yüzeyde herhangi bir bozulma yaratmadığı kabulünü yapmaktır. Böyle bir yaklaşım matematiksel olarak ancak Froude sayısının sıfır olması durumunda doğrulanabilir. Bu durumda serbest yüzey sınır koşulu $\Phi_z = 0$ halini alır ve double model gösterimi gerçekleşmiş olur. Double model halinde cisim ve onun serbest yüzeye göre simetrik sanal ikizi oluşturulmuş olunur. Bu problemin çözümü bir önceki probleme göre daha kolay ve çabuk elde edilir. Denklem (2.10) bu problem içinde geçerliliğini devam ettirmektedir.

$$\frac{1}{4\pi} \iint_S \sigma(\xi,\eta,\zeta) G_n(x,y,z;\xi,\eta,\zeta) dS(\xi,\eta,\zeta) = -V_o n_x(x,y,z) \qquad (2.14)$$

Sadece Green fonksiyonunun ifadesi değişmiş bulunmaktadır. Green fonksiyon ifadelerini yeniden yazarsak;

Büyük Froude sayıları için;

$$G(x,y,z;\xi,\eta,\zeta) = \frac{1}{\sqrt{a^2+b^2+c^2}} - \frac{1}{\sqrt{a^2+b^2+\overline{c}^2}} \qquad (2.15)$$

Sıfıra yaklaşan Froude sayıları için;

$$G(x,y,z;\xi,\eta,\zeta) = \frac{1}{\sqrt{a^2+b^2+c^2}} + \frac{1}{\sqrt{a^2+b^2+\overline{c}^2}} \qquad (2.16)$$

burada;

$a = x - \xi \qquad b = y - \eta \qquad c = z - \zeta \qquad \overline{c} = z + \zeta$

Double model yaklaşımı için (2.16) denklemindeki Green fonksiyonu kabul edilmiştir.

(2.14) denkleminin çözümü, geçtiğimiz son zamanlarda detaylı olarak incelenmiş ve hesap zamanının azaltılması ve de cismi temsil eden panel sayısının arttırılması üzerinde çalışılmıştır. Bilgisayar kapasitelerinin artması ile detaylı işlemlerin yapılabilmesi olanaklı hale gelmiştir.

Bir diğer önemli katkı ise yüksek dereceden etkilerin göz önüne alınması ile sağlanmıştır. Hess ve Smith (1967) kaynak şiddetlerinin panel üzerindeki dağılımının cismin yüzey eğriliğine bağlı olduğunu göstermiştir. Bu çalışmalar ayrıca panel geometrilerinin ve üzerlerindeki kaynak şiddeti dağılımlarının olumsuz yönlerine de değinmişlerdir.

3. ÇÖZÜM YÖNTEMİ

Problemi analitik olarak çözmek gemi yüzeyinin karmaşıklığı nedeni ile olanaksız olduğundan sayısal çözüm aramak zorundayız. Bu nedenle geminin S yüzeyini, N yeteri kadar büyük olmak üzere, N tane $\{S_n\}$ yüzey elemanına böleceğiz Yüzeylerin panelleme işlemi Ek 2'de detaylı olarak açıklanacaktır. Böylece elde edilen yüzey elemanları yeteri kadar küçük olacağı için, bunların her biri üzerinde kaynak şiddeti sabit varsayılabilir ve (2.10) integral denklemi aşağıdaki gibi yazabiliriz.

$$\sum_{n=1}^{N}\sigma_n \iint_{S_n} G_n(x,y,z;\xi,\eta,\zeta)dS(\xi,\eta,\zeta) = -V_o n_x \tag{3.1}$$

Burada (x,y,z) noktasının gemi yüzeyi üzerindeki m'inci yüzey elemanına ait alan merkezi olduğunu ve her yüzey elemanı üzerinde kaynak şiddetinin sabit olduğunu varsayalım. Gemi yüzeyi sınır koşulunun her bir yüzey elemanının alan merkezinde sağlandığı düşünür ve her bir yüzey elemanı üzerindeki entegrasyonu

$$\iint_{S_n} G_n(\xi_m,\eta_m,\zeta_m;\xi,\eta,\zeta)dS(\xi,\eta,\zeta) = G_n(\xi_m,\eta_m,\zeta_m;\xi_n,\eta_n,\zeta_n)\Delta S_n = G^n_{mn} \tag{3.2}$$

şeklinde tanımlarsak 3.1 entegral denklemi

$$\sum_{m=1}^{N} G^n_{mn}\sigma_n = -V_o n_{xm} \qquad m = 1 \Rightarrow N \tag{3.3}$$

lineer denklem sistemine indirgenir.

Kelvin kaynağının türevlerinin yüzey elemanları üzerindeki değerini sayısal olarak hesaplamak için kaynağın türevlerini üç bölümde göz önüne almak gereklidir. Birinci bölümde Rankine kaynağı ve bunun imajından oluşan kısmın hesabını ele alacağız. Bu terimlerin hesabında tek sorun Rankine kaynağının tekilliği nedeni ile çıkar ve kaynağın kendisi üzerindeki etkisi yüzey elemanı üzerindeki entegrasyonun limit işlemi ile yapılması sonucu belirlenir. Limit işleminin uygulanışı Ek 1'de verilmiştir (Okan, 2000). Ayrıca kaynağın kendisi üzerindeki etkisinde var olması muhtemel asimetriyi de göz önüne almak ve de sayısal entegrasyonun hassasiyetini artırmak için her yüzey elemanı dört paçaya bölünerek entegrasyon yapılır.

İkinci olarak tek katlı entegralin her bir panel üzerinde değerlendirilmesini göz önüne alcağız. Tek katlı entegralin sayısal hesabı genelde herhangi bir problem çıkarmamakla birlikte integrandın yapısı nedeni ile bazı hallerde dikkatli davranmak gerekir. Sorun integranddaki

$Sec\theta$ teriminin θ açısının $\pi/2$ değerine yaklaşırken sonsuz büyüklüğe ulaşmasından kaynaklanmaktadır. Tek katlı integraldeki integrant çok fazla salınım yapar ve üstel sönümün az olduğu yerlerde bu salınımlar sayısal hesabın yeterli hassasiyette olabilmesini engelleyebilir. Bu sorunu çözebilmek amacı ile 'stasyoner faz' yöntemine dayalı bir sayısal entegrasyon yöntemi geliştirildi. Bu yöntemin ayrıntıları ve uygulanış şekli Ek 2'de verilmektedir (Okan, 2000).

Son olarak da iki katlı entegralin irdelenmesine değineceğiz. İki katlı entegraldeki zorluk dalga sayısının $\kappa_0 Sec^2\theta$ değerine yaklaşırken tekillik göstermesi sonucu ortaya çıkmaktadır. Bu sorunu çözebilmek için κ üzerindeki entegrali, $\kappa^* >> \kappa_0 Sec^2\theta$ olmak üzere, 0 ile κ^* arasında ve κ^* ile ∞ arasında iki entegrale ayırarak hesaplayacağız. Birinci entegralde düğüm noktalarını, entegrasyon aralığını eşit aralıklara bölüp tekilliği de herhangi iki düğüm noktasının tam ortasına gelecek şekilde seçersek, entegralin sayısal olarak hesaplanmasında hiç bir sorunla karşılaşılmaz. İkinci entegralde κ^* değerini yeteri kadar büyük seçtiğimiz için

$$\frac{\kappa}{\kappa - \kappa_0 Sec^2\theta} \approx 1$$

olacağı açıktır. Bu durumda entegral basitleşir ve analitik olarak hesaplamak mümkün olur.

İki katlı entegralin hesabında kaynaktan uzak noktalarda radyasyon koşulu göz önüne alınarak basitleştirme yapmak olanaklıdır. Radyasyon koşulu gereği iki katlı entegralin mutlak değeri kaynaktan uzakta tek katlı entegrale yaklaşır. Kaynağın ön tarafında işaretleri ters olduğu için iki entegralin toplamı sıfır olur. Kaynağın gerisinde ise işaretleri aynıdır ve iki katlı entegral yerine tek katlı entegralin değeri kullanılabilir.

Rankin kaynağının türevlerinin alan üzerinde entegrasyonunda, hesap hassasiyetini kaybetmeksizin tekillikten kurtulmak amacı ile, alan elemanlarını dörde bölerek yapılmıştı. Dalga yaratılmasına karşı gelen son iki terimin hesabında, bu terimlerin alan elemanları üzerindeki entegrasyonunda tekillik söz konusu olmadığı ve entegrasyon çok daha fazla zaman aldığı için, alan elemanlarını dörde bölme işlemini sadece sınır koşulunu sağladığımız noktaya çok yakın olan alan elemanlarında uygulamak yeterli olur.

Gemi yüzey elemanlarının seçimi sonuçların hassasiyeti açısından önemli rol oynar. Teorik olarak yüzey elemanlarının sayısını büyüterek çözümü analitik çözüme yaklaştırmak mümkündür ancak bu hem pratik değildir hem de sayısal hesabın stabilitesini olumsuz yönde etkileyebilir. Yüzey elemanlarının sayısını her bir elemanın yeteri kadar küçük olmasını sağlayacak kadar büyük seçmeli ancak sayısal hesabın stabilitesini olumsuz etkilemekten kaçınmalıdır. Ayrıca yüzey elemanlarının seçiminde bu elemanların alanlarının birbirinden çok farklı olmamasına, ince uzun ve üçgene yaklaşan elemanlardan kaçınmaya özen gösterilmelidir.

4. DALGA DİRENCİNİN HESABI

Gemilerin dalga direnci ilerlemekte olan gemiye etki eden potansiyel kuvvetlerin geminin ilerleme doğrultusundaki bileşeni olup bunu iki yoldan hesaplamak mümkündür. Birinci yöntem gemiye etki eden kuvvetlerin gemi yüzeyindeki dinamik basınç dağılımını geminin dış normali yönünde entegrali olacağı ilkesine dayanır. Bu durumda geminin ilerleme yönü x doğrultusu olduğundan

$$R = \iint_S n_x (p - p_\infty) dS \tag{4.1}$$

şeklinde tanımlanabilir. Diğer taraftan C_p basınç katsayısı ile C_w dalga direnci katsayılarının

$$C_p = \frac{(p - p_\infty)}{\frac{1}{2}\rho V_o^2 S} \tag{4.2}$$

$$C_w = \frac{R}{\frac{1}{2}\rho V_o^2 S} \tag{4.3}$$

olarak tanımlandığı göz önüne alınırsa dalga direnci katsayısı için

$$C_w = \iint_S n_x C_p dS \tag{4.4}$$

yazılabilir. Daha evvelce her yüzey elemanı üzerinde basınç katsayıları hesaplanmış olacağından dalga direncinin hesabı

$$C_w = \sum_{n=1}^{N} n_{xn} C_{pn} S_n \tag{4.5}$$

toplamının hesabına indirgenir.

İkinci yöntem momentumun korunumu ilkesine ve geminin yeteri kadar ilerisinde herhangi bir akım oluşmayacağı koşuluna dayanır. Bu durumda gemi direnci formülü;

$$R = \rho \int_o^\infty \int_o^\infty \left[u^2 - v^2 - w^2\right] dy dz - \rho g \int_o^\infty \zeta^2 dy \tag{4.6}$$

şeklinde verilebilir (Wehausen, 1975). Öte yandan geminin yeteri kadar gerisinde serbest su yüzeyini

$$\zeta(x,y)=\int_{0}^{\frac{\pi}{2}}\left[a(\theta)\mathrm{Cos}\alpha(\theta)x+b(\theta)\mathrm{Sin}\alpha(\theta)x\right]\mathrm{Cos}\beta(\theta)y\,d\theta \tag{4.7}$$

şeklinde yazabileceğimizi ve bu durumda potansiyeli

$$\phi(x,y,z)=V_{o}\int_{0}^{\frac{\pi}{2}}e^{-vz\mathrm{Sec}^{2}\theta}\left[a(\theta)\mathrm{Sin}\alpha(\theta)x-b(\theta)\mathrm{Cos}\alpha(\theta)x\right]\mathrm{Cos}\beta(\theta)y\,d\theta \tag{4.8}$$

olarak göstermek mümkündür. Bu tanımlar (4.2) denklemine yerleştirilir,

$$\begin{aligned} f(x)&=\frac{2}{\pi}\int_{0}^{\infty}F(\theta)\mathrm{Cos}x\theta\,d\theta &&\Rightarrow &&\int_{0}^{\infty}\{f(x)\}^{2}dx=\frac{2}{\pi}\int_{0}^{\infty}\{F(\theta)\}^{2}d\theta\\ g(x)&=\frac{2}{\pi}\int_{0}^{\infty}G(\theta)\mathrm{Sin}x\theta\,d\theta &&\Rightarrow &&\int_{0}^{\infty}\{g(x)\}^{2}dx=\frac{2}{\pi}\int_{0}^{\infty}\{G(\theta)\}^{2}d\theta \end{aligned} \tag{4.9}$$

ilişkilerinden yararlanılır ve dalga direnci katsayısının tanımı kullanılırsa dalga direnci katsayısı

$$C_{w}=\frac{\pi}{2S}\int_{0}^{\frac{\pi}{2}}\left[a^{2}+b^{2}\right]\mathrm{Cos}^{3}\theta\,d\theta \tag{4.10}$$

şeklinde bulunur. Burada $a(\theta)$ ve $b(\theta)$ katsayılarını serbest su yüzeyinin Fourier transformlarını alarak buluruz.

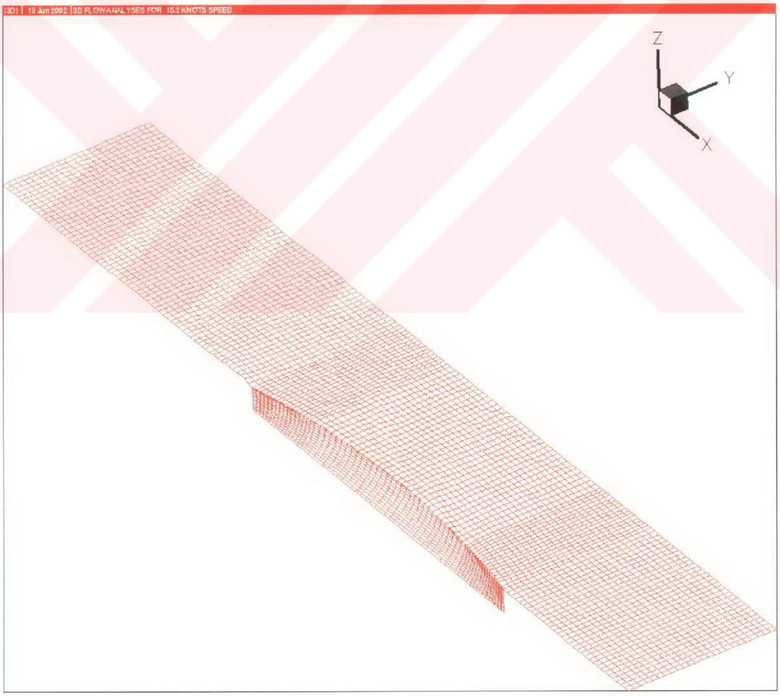
5. SAYISAL UYGULAMALAR

5.1 Wigley Formu – Serbest Yüzeyli Analiz

İlk uygulama wigley gemi formu olarak bilinen model üzerinde yapılmıştır. Burada form hem x doğrultusunda hem de z doğrultusunda kesitleri parabol olan bir formdur ve

$$y(x,z)=\frac{B}{2}\left[1-\left(\frac{2x-L}{L}\right)^{2}\right]\left[1-\left(\frac{z}{d}\right)^{2}\right] \tag{5.1}$$

formülü ile verilmektedir. Bu formülde L geminin su hattı boyu 100 m, B kalıp genişliği 10 m ve d de su çekimi olan 6.25 m'yi göstermektedir. Gemi boyuna 72 posta ile ve her posta da 16 nokta ile tanımlanmış böylece gemi yüzeyi yaklaşık 1160 elemandan ve serbest yüzey yaklaşık 5000 elemandan oluşmuştur. Formun yüzey panelleme sistemi Şekil 5.1'de görülebilmektedir.

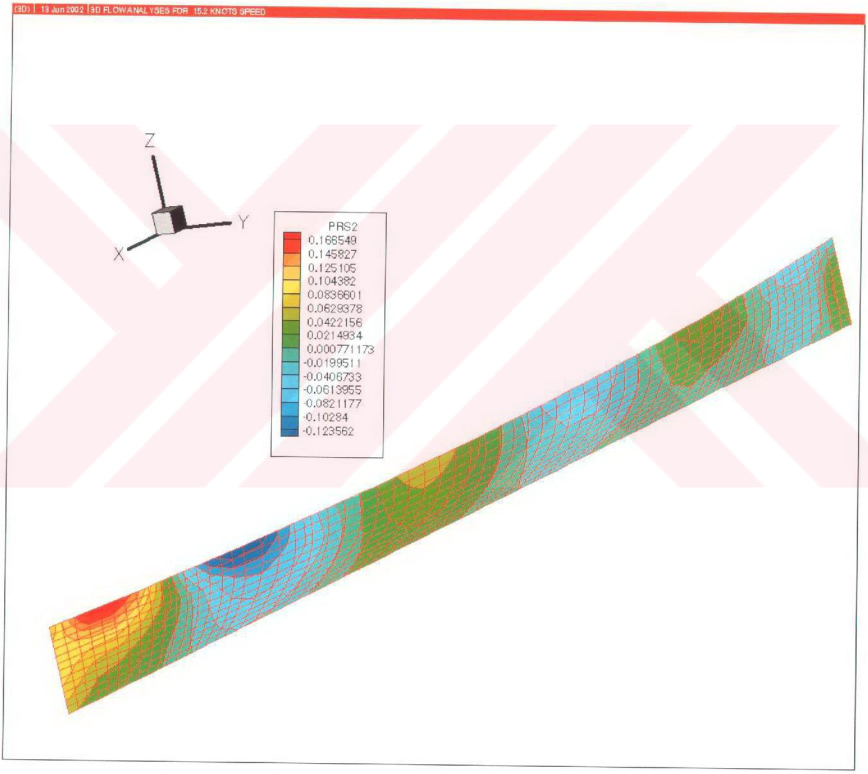


Şekil 5.1 – Wigley formunun yüzey panelleme sistemi

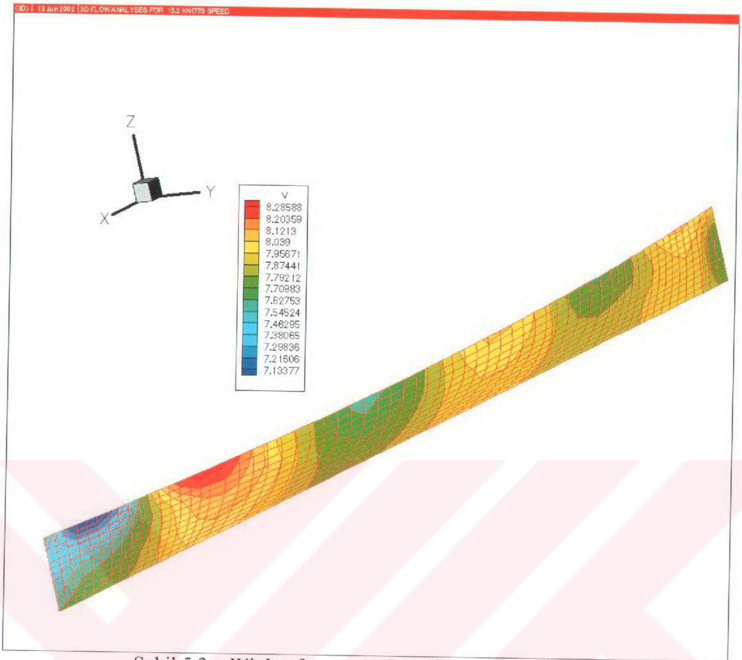
Oluşturulan panel sistemi program yardımı ile Froude sayısı 0.25 için çözdürülmüştür ve serbest yüzey değişimi, form etrafındaki hızlar ve basınçlar sonraki resimlerde verilmiştir.

Serbest su yüzeyine yaklaştıkça dinamik basıncın artacağı, geminin dibine doğru basıncın azalacağı göz önüne alınırsa elde edilen sonuçlar gerçeğe yakın gözükmektedir.

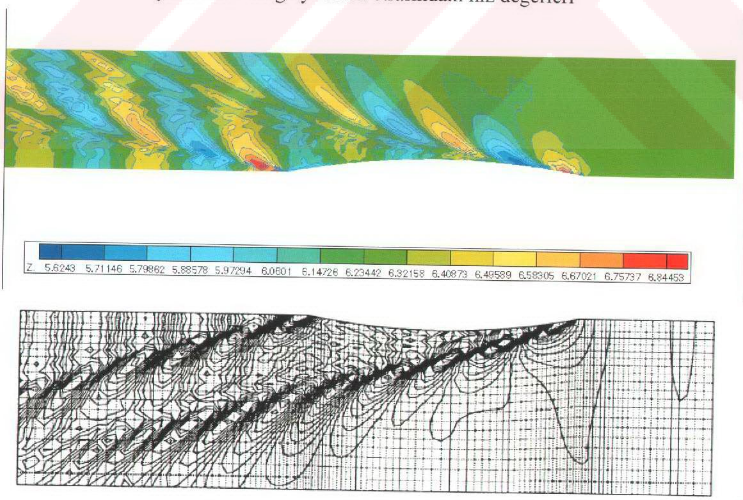
Şekil 5.4'de bulunan serbest yüzey değişim değerleri yapılan deneyler ile karşılaştırılarak verilmiştir ve oluşan dalga profilinin benzerliği görülmüştür (Inui, 1976).



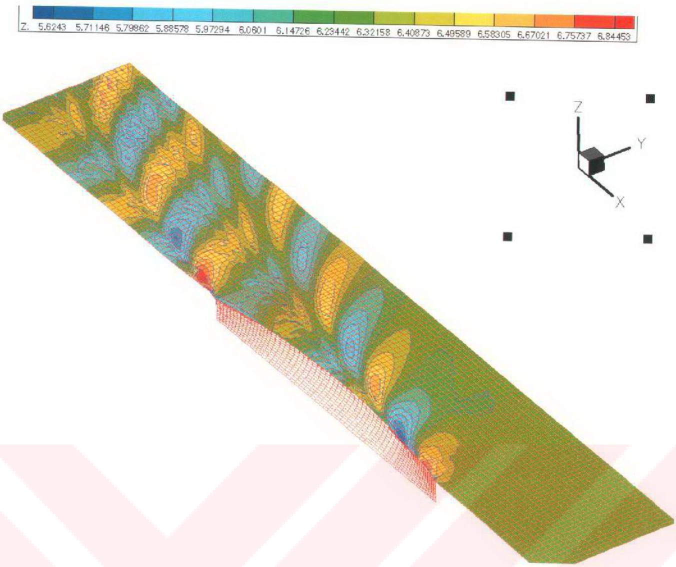
Şekil 5.2 – Wigley formu etrafındaki basınç değerleri



Şekil 5.3 – Wigley formu etrafındaki hız değerleri



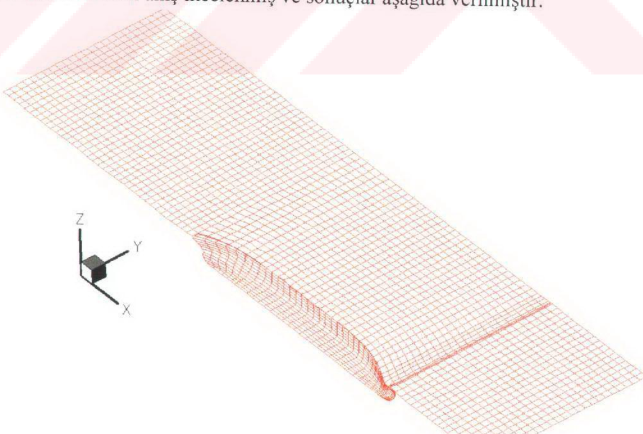
Şekil 5.4 – Wigley formu serbest su yüzeyindeki değişim



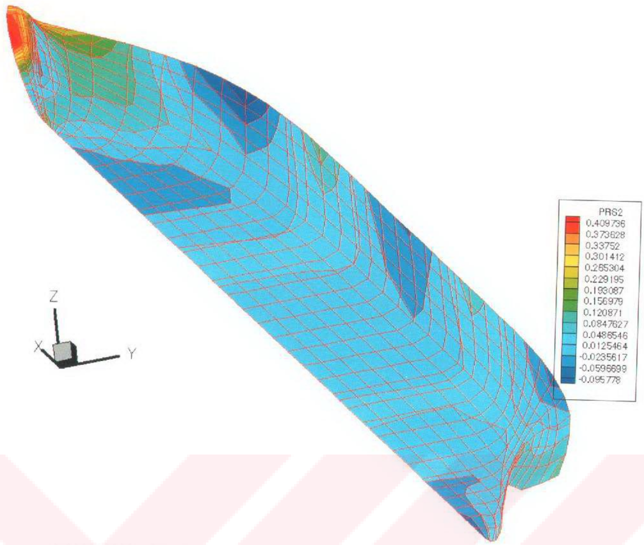
Şekil 5.5 - Wigley formu serbest su yüzeyindeki değişim

5.2 Tanker Formu (5850 Dwt) – Serbest Yüzeyli Analiz

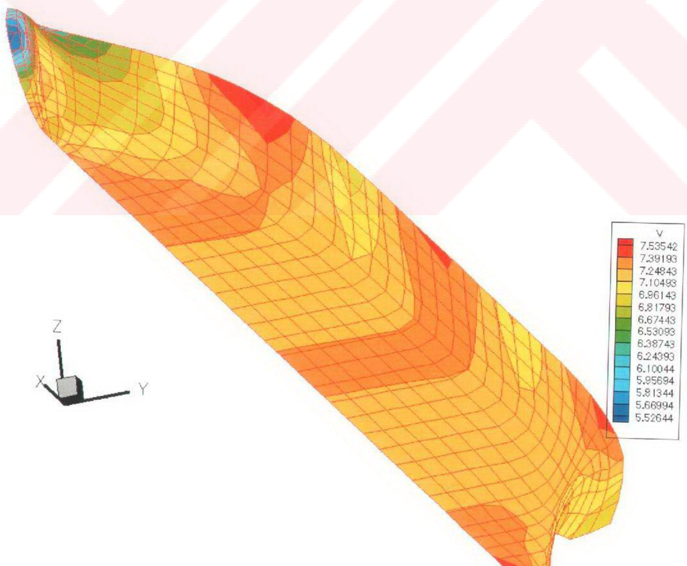
Delta Denizcilik ve Mühendislik tarafından dizayn edilmiş 5850 Dwt kapasiteli bir kimyasal tankerin formu etrafındaki akış incelenmiş ve sonuçlar aşağıda verilmiştir.



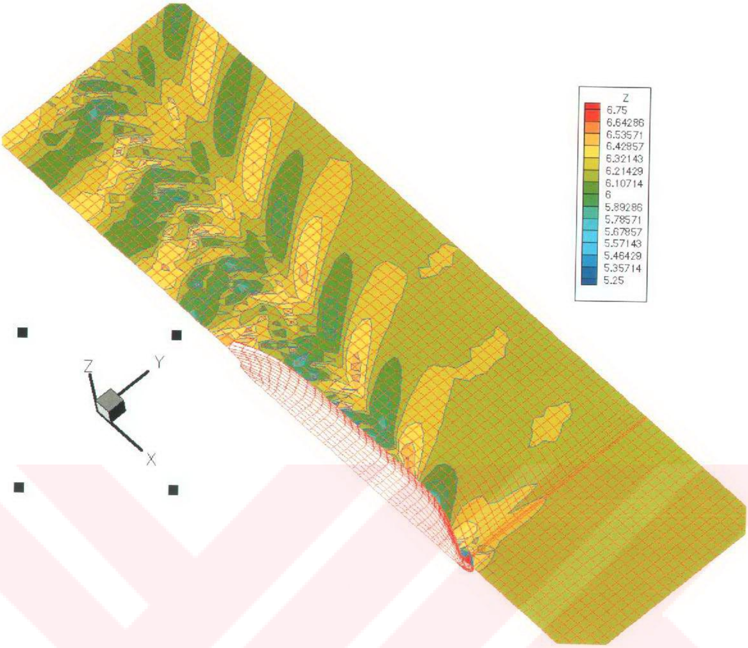
Şekil 5.6 – Tanker formunun panelleme sistemi



Şekil 5.7 – Tanker formu etrafındaki basınç dağılımı

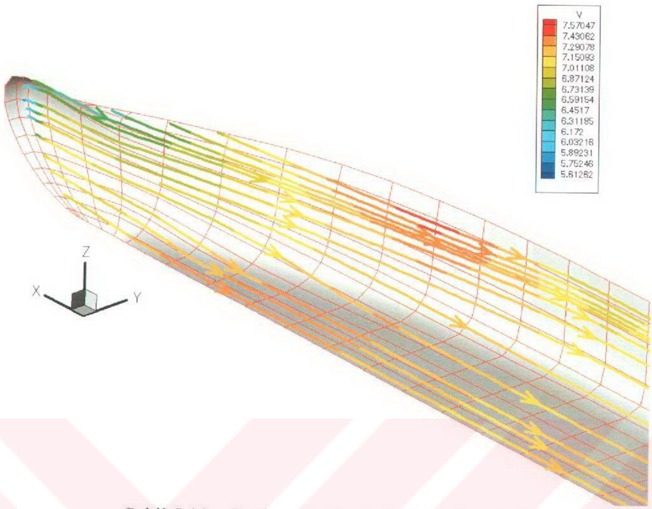


Şekil 5.8 – Tanker formu etrafındaki hız dağılımı



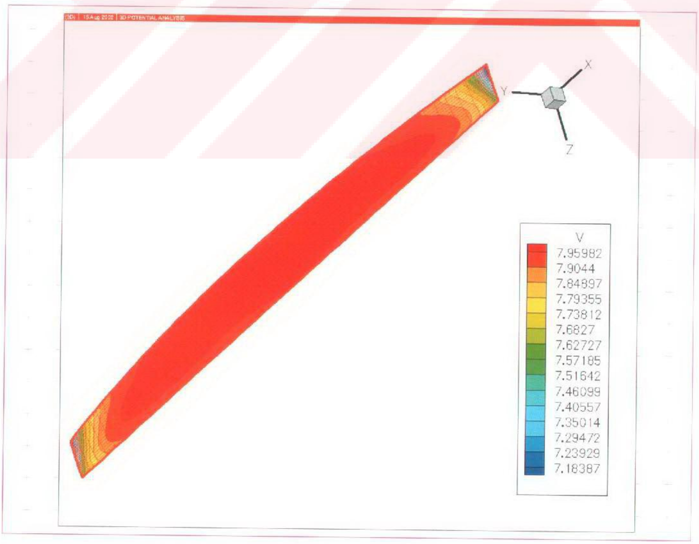
Şekil 5.9 – Tanker formu serbest su yüzeyi değişimi

Şekil 5.10 – Tanker formu serbest su yüzeyi değişimi

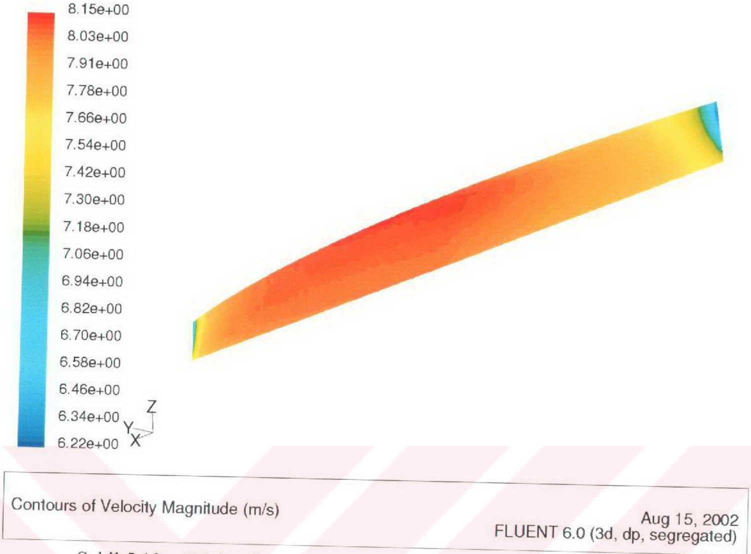


Şekil 5.11 – Tanker formu etrafındaki akım hatları

5.3 Wigley Formu – Serbest Yüzeysiz Analiz

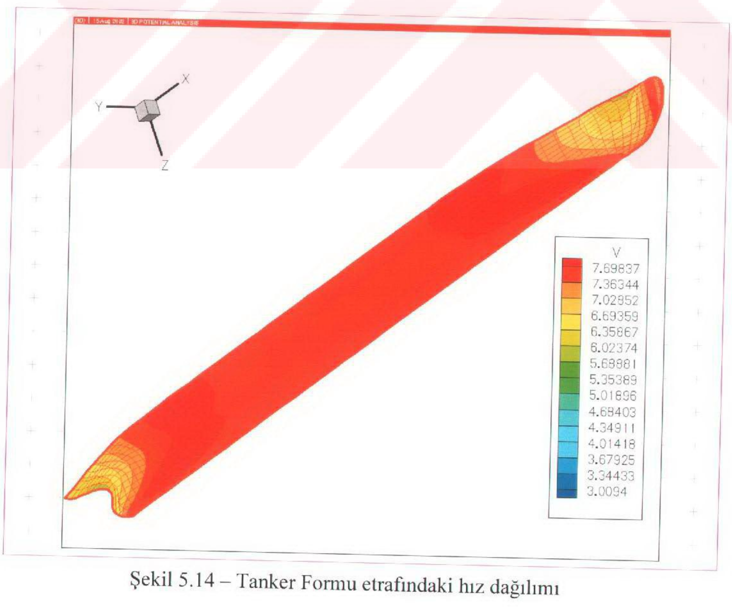


Şekil 5.12 – Wigley formu etrafındaki hız dağılımı

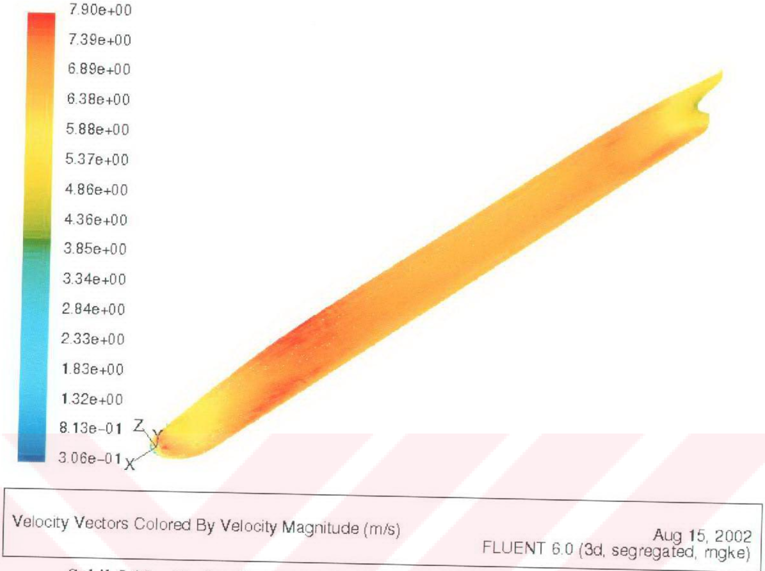


Şekil 5.13 – Wigley formu etrafındaki hız dağılımı (Fluent Sonucları)

5.4 Tanker Formu (6750 Dwt) – Serbest Yüzeysiz Analiz



Şekil 5.14 – Tanker Formu etrafındaki hız dağılımı



Şekil 5.15 – Tanker Formu etrafındaki hız dağılımı (Fluent Sonucu)

6. SONUÇLAR

Gemi etrafındaki potansiyel akımı, hızları, basınç katsayılarını ve bunlardan dolayı ortaya çıkan dalga direncini hesaplamak amacı ile bir program yazılmış ve wigley formu ve bir tanker formu örneğine uygulanmıştır. Ayrıca sebest su yüzeyi gözardı edilerek çözümler yapılmış ve Delta Marine lisanslı Fluent 6.0 programı sonucları ile karşılaştırılmıştır. Uygulamaların sonuçları incelendiğinde şu genel gözlemleri yapabiliriz.

Elde edilen sonuçlar, ön dizayn aşamasında gemi sualtı formunun bozuk olduğu bölgelerin belirlenmesinde ve bu bölgelerde formun düzeltilmesinde basınç dağılımı ve hız vektörlerinden yararlanılabilir.

Sonuçlarda bazı yüksek frekanslı salınımlar gözlenmiş olup bunların Green fonksiyonlarının entegrasyonundan kaynaklandığı tahmin edilmektedir.

Diğer taraftan programı geliştirmek için de bazı adımlar atmak olanaklıdır. İlk etapta yapılabilecek geliştirmeleri şöyle sıralayabiliriz.

Yukarıda anlatılan yönteme bir integral sınır tabaka hesabı eklemek sureti ile gemi yüzeyinde kayma gerilmesini ve dolayısıyla geminin sürtünme direncini hesaplamak olanağı vardır.

Viskozitenin potansiyel akım ve dolayısıyla kaynak dağılımı üzerindeki etkisi büyük bir olasılıkla sınır tabakanın kalınlığı ile yakından ilişkilidir. Bu göz önüne alındığında hesaplanmış olan sınır tabaka kalınlığına bağlı olarak gemi kıç tarafındaki kaynaklar için sözü edilen katsayıyı verecek ampirik bir yöntem geliştirerek dalga direncinin daha hassas olarak hesaplanması olanaklıdır.

Teoride kullanılan, Green fonksiyonu içerisindeki lineerize edilmiş serbest su yüzeyi formülasyonu yerine, gerçek su yüzeyi sınır koşullarını sağlayan formülasyon kullanılarak bulunan dalga yüksekliklerinin kontrolü gerekmektedir.

Potansiyel teori sonucları, RNG Türbülans modeli kullanılarak elde edilen Fluent sonuçları ile karşılaştırılmıştır ve kabul edilebilir sınırlar içinde kaldığı sonucuna varılmıştır.

KAYNAKLAR

Aldoğan, A. İ., (1977)‘ Lineer olmayan dalga direnci teorisi ve uygulaması’, İ.T.Ü. Gemi İnşaatı Fakültesi Doktora Tezi, İstanbul.

Havelock, T. H., (1966) ‘ The collected papers of Sir Thomas Havelock on hydrodynamics’, Edited by C. Wigley, Office of Naval Research, Washington D.C.

Hess, J. L. ve Smith, A. M. O., (1967)‘Calculation of potential flow about arbitrary bodies’, Paper No. 1 ‘Progress in aeronautical sciences, Vol. 8’, Edied by D. Kucheman, Pergamon Press,

Inui T. ,(1976) ‘ Introductory remarks ’ International seminar on wave resistance, University of Tokyo, Japan: 7-18

Okan B.M. Gül Y., (2000) ‘Düşük ve orta hızlı gemilerin dalga direncinin hesabı için sayısal bir yöntem’, Yayınlanmamış Rapor, Universty of Glasgow.

Sabuncu, T., (1962) ‘Gemilerin dalga direnci teorisi’, İstanbul Teknik Üniversitesi Gemi Enstitüsü Bülteni 12, İstanbul.

Sabuncu, T., (2000) ‘Free Surface Hydrodynamics’, İstanbul Teknik Üniversitesi Yayınları , İstanbul.

Şaylan, Ö., (1980) ‘Potential Flow Around Three-Dimensional Arbitrary Bodies ’, Gemi Enstitüsü Bülteni No 24, İstanbul.

Wehausen, J. V., (1975) ‘ The Wave Resistance of Ships’, Advances in Applied Mechanics Vol 13, Academic Press.

EKLER

Ek 1 : Green Fonksiyonunun Türevlerindeki Tekilliğin Limit Halde Hesabı

Green fonksiyonundaki ilk terim kontrol noktası ile kaynak noktasının çakışması halinde tekillik gösterdiği için hem sayısal hesap açısından hem de tekilliğin değerlendirilmesi açısından sorun çıkarır. Sayısal açıdan sorun her bir yüzey elemanını dört parçaya bölmek sureti ile kolayca çözülür çünkü kaynak noktasının kontrol noktası ile çakışma olasılığı ortadan kalkmış olur. Ne var ki bu tekillikten dolayı gelecek katkıyı doğru olarak hesaplamaya olanak vermez. Bu hesabı doğru olarak yapabilmek için analitik olarak limit işlemini göz önüne almak gerekir.

Bu şekilde P noktası bir kontrol noktası Q noktası da bir kaynak noktası olup aralarındaki küçük uzaklık ortadan kalktığında limit hale ulaşmış oluruz. Burada kaynak dağılımının çok küçük ε yarıçaplı bir daire üzerinde olduğunu ve bu daire içerisinde kaynak şiddetinin sabit olacağını düşünürsek potansiyele Green fonksiyonundaki tekillikten gelecek katkının

$$\nabla\phi = \lim_{P \to Q} \left\{ \frac{\sigma_P}{4\pi} \int_0^{2\pi} d\theta \int_0^{\varepsilon} \nabla \left[\frac{1}{|\vec{r}(PQ)|} \right] \rho d\rho \right\}$$

olacağı açıktır. Bu denklemde $\vec{r}(p,Q)$ kaynak ile kontrol noktası arasındaki yer vektörü olup yüzey normali ve iki teğet doğrultusu cinsinden

$$\vec{r}(P,Q) = \rho(\mathrm{Cos}\theta\vec{t} + \mathrm{Sin}\theta\vec{s}) + h\vec{n}$$

şeklinde yazılabileceği ve

$$\nabla \frac{1}{|\vec{r}(P,Q)|} = \frac{-\vec{r}(P,Q)}{\left[|\vec{r}(P,Q)|\right]^3}$$

olduğu göz önüne alındığında ve θ üzerinde 0 ile 2π arasında Cosθ ve Sinθ fonksiyonlarının entegrasyonunun sıfır olacağı hatırlanırsa Ek I.1 denklemini şu şekilde yazmak olanağı vardır.

$$\nabla\phi = \lim_{h,\varepsilon \to 0} \left\{ \frac{\sigma_P}{2} \int_0^{\varepsilon} \frac{-\rho h\vec{n}}{\left[\sqrt{\rho^2 + h^2}\right]^3} d\rho \right\}$$

Böylece limit halinde hız bileşenlerine Green fonksiyonundaki tekillikten katkının

$$\nabla\phi = \lim_{h,\varepsilon \to 0} \frac{\sigma_P h\vec{n}}{2} \left[\frac{1}{\sqrt{\varepsilon^2 + h^2}} - \frac{1}{h} \right] = -\frac{\sigma_P}{2}$$

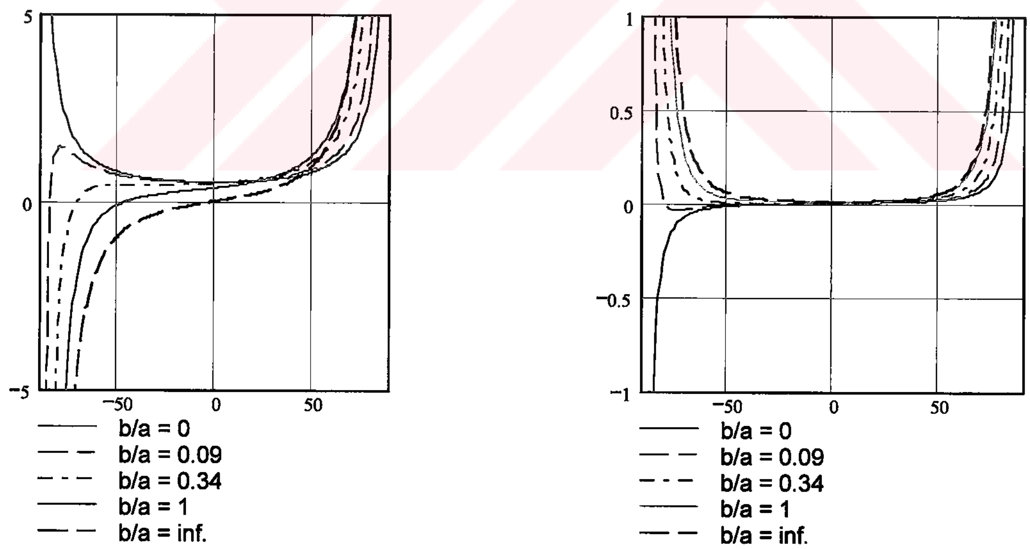
olduğu bulunur. Burada limit halinde sonuç daire yarıçapı ε'dan bağımsız gibi gözükmekle birlikte kaynak dağılımını sabit varsayabilmek için bu yarıçapın çok küçük olması gerektiğini unutmamak gerekir.

Ek 2 : Stasyoner Faz Yönteminin Sayısal Entegrasyona Uygulanışı

Green fonksiyonundaki tek katlı entegral θ'nın büyük değerlere ulaşması ile çok fazla salınım yapmaya başlar. Bu durumda tek katlı entegralin türevleri

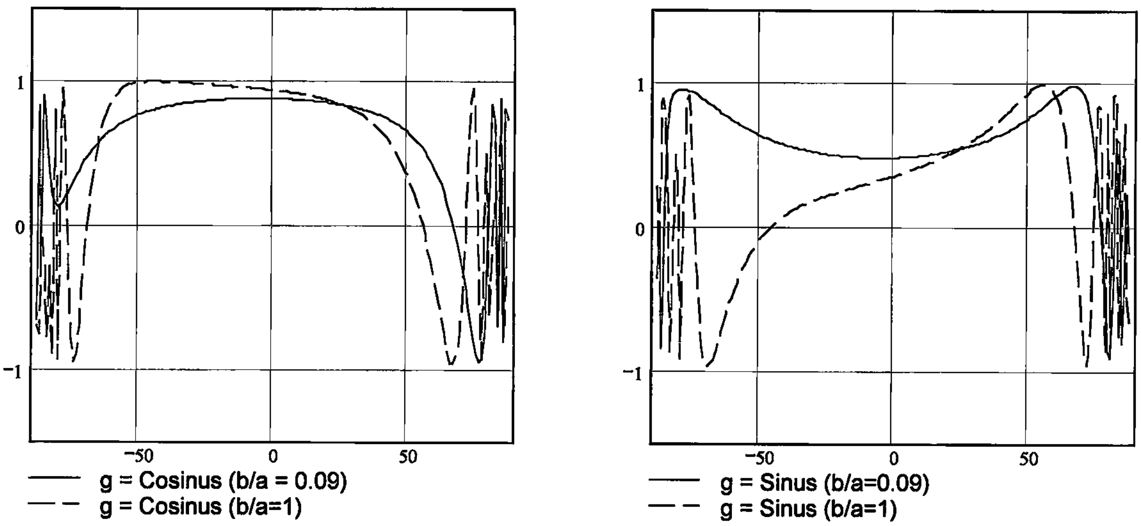
$$I=\frac{\kappa_o}{2\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} f(\kappa_o,\theta)g[\varphi(a,b,\kappa_o,\theta)]d\theta$$

şeklinde yazılabilir. Burada f fonksiyonu da x, y ve z kısmi türevleri için sırası ile $e^{-\kappa_o\bar{c}Sec^2\theta}Sec^3\theta$, $e^{-\kappa_o\bar{c}Sec^2\theta}Sec^4\theta Sin\theta$ ve $e^{-\kappa_o\bar{c}Sec^2\theta}Sec^4\theta$ şeklinde verilir ve yavaş da olsa salınım yapmadan sönerler. Diğer taraftan g fonksiyonu x ve y türevleri için cosinüs φ z türevi için de sinüs φ olup entegralin esas salınım yapan kısmını oluşturur. Bu yapıdaki φ faz fonksiyonu adını alır ve $\varphi=\kappa_o Sec^2\theta(aCos\theta+bSin\theta)$ şeklinde tanımlanmaktadır. Faz fonksiyonu a ve b büyüklüklerinin oranına bağlı olarak değişmekte olup büyük değerlere erişmesi ile entegrantın aşırı salınım yapmasına neden olur. Stasyoner faz yöntemi bu entegrale önemsenir katkının sadece faz fonksiyonunda değişimin çok yavaş olduğu bölgelerde olacağı ilkesinden yararlanır. Şekil Ek2.1 faz fonksiyonunun ve türevinin çeşitli b/a oranları için vermektedir.



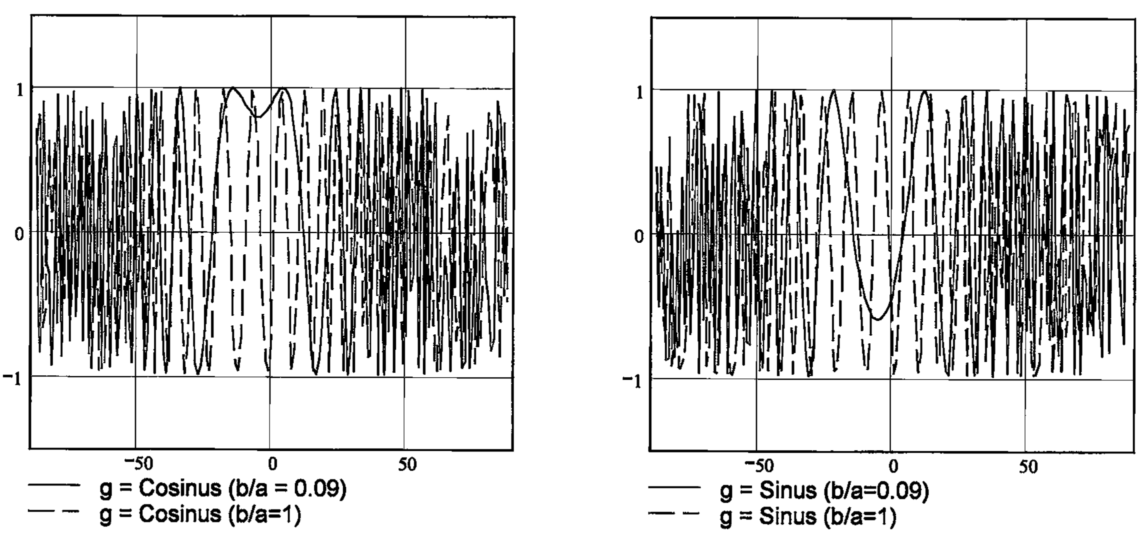
Şekil Ek2.1 a : Faz fonksiyonunun değişimi b : Faz fonksiyonunun türevinin değişimi

Yukarıdaki grafikler incelendiğinde integrantların ± 90 dereceye yaklaştıkça aşırı salınım yapacağı bunun dışında oldukça yavaş değişeceği açıktır. Şekil Ek2.2 kaynağa yakın durumda ve yavaş sönümün olduğu hallerde g fonksiyonunun sınüs veya cosinüs karakterindeki değişimini göstermektedir.



Şekil Ek2.2 : Green fonksiyonu türevlerindeki entegrantların kaynağa yakın noktalar için değişimleri

Bu durumda sayısal entegrasyon yapılırken entegrasyon aralığını $-\pi/2$ ile $\pi/2$ arasında değiştirmek ile birlikte faz fonksiyonunun türevinin mutlak değeri 0.8'i aştığı aralıklarda entegrasyonu ihmal etmek gerekmektedir. Kaynaktan uzaklaştıkça ve özellikle b/a oranı arttıkça entegrasyon aralığı kısalır. Buna karşı gelen bir örnek kaynak ile kontrol noktası arasındaki mesafenin 8 katı artması hali için Şekil Ek2.3'de verilmektedir.

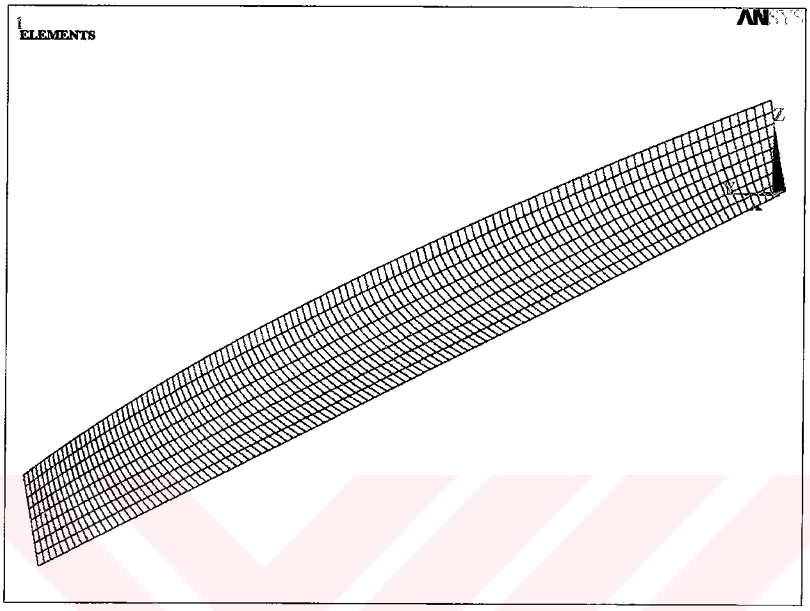


Şekil Ek2.3 : Green fonksiyonu türevlerindeki entegrantların kaynaktan uzak noktalar için değişimleri

Yukarıdaki şekilden de görüldüğü gibi b/a = 1 için entegrasyon her iki halde de sıfır vermesi gerekir. Bu gemiden çok uzakta ve orta simetri ekseni ile Kelvin açısı olarak bilinen 19.47^{o} açı yapan doğrultu dışında herhangi bir şekilde dalga oluşmayacağını gösterir.

Ek 3 : Gemi Yüzeyi Panelleri İle İlgili İşlemler

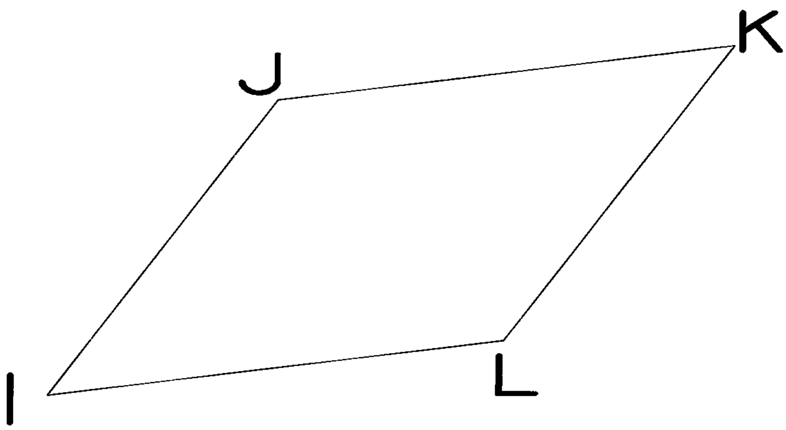
Gemi yüzeyinin panelleri Delta Marine lisanslı Ansys 6.0 genel amaçlı sonlu elemanlar programında elde edilmiştir ve yazılan bir macro ile akış problemi için yazılan programa import edilmiştir. Gemi yüzey yüzeyi panelleri için bir örnek Sekil Ek3.1' de görülebilir.



Ek3.1 –Gemi yüzey panelleri

Elde edilen panel sisteminin programa import edilmesi, macro yardımıyla Ansys programından bir dosyaya yazdırılan panellerin nokta numaralarının ve bu noktaların koordinatlarının okunması ile gerçekleştirilmiştir.

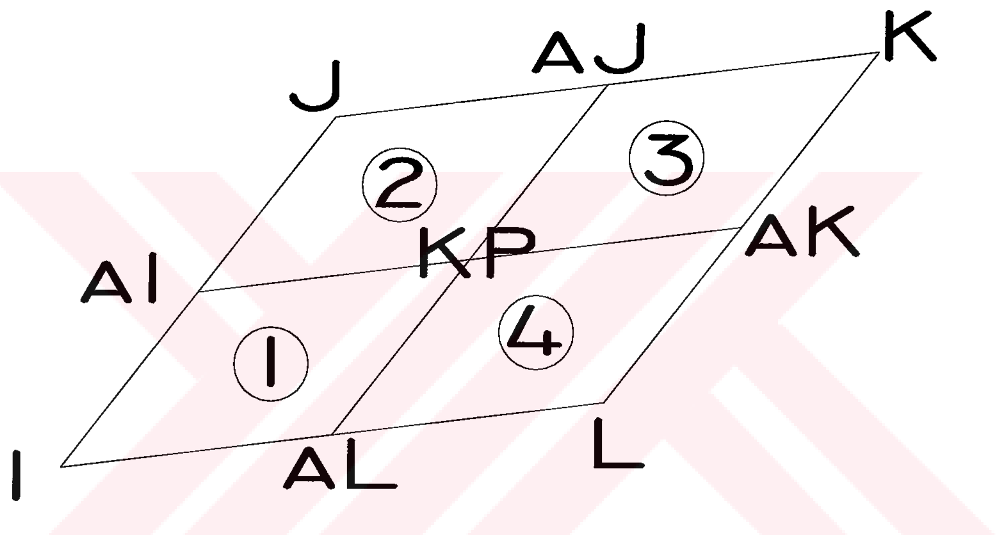
Akış programı içerisinde ilkönce, okutulan panellerin köşe noktalarının koordinatları yardımı ile her bir panelin alanı, kontrol noktası ve normal doğrultu kosinüslerinin hesabı yapılmıştır. Bir panelin köşe noktalarını i, j, k ve l harfleri ile (Şekil Ek3.2) tanımladığımızı düşünelim.



Şekil Ek3.2 – Panel tanımlama sistemi

Panel alanları ijl ve jkl üçgenlerinin alanlarının hesaplanıp toplanması ile elde edilmiştir. Panel kontrol noktası ise i, j, k, ve l noktalarının koordinatlarının toplanıp 4'e bölünmesi ile hesaplanmıştır. Panellerin normal doğrultu kosinüslerinin hesaplanması için ij ve il vektörlerinin vektörel çarpımları bulunarak birim vektör cinsinden açılar bulunmuştur.

Panellerin kendi üzerlerindeki etkilerin hesaplamak amacı ile paneli 4'e bölüp alt paneller elde edildiği açıklanmıştı. Paneller Şekil Ek3.3 'te gösterildiği gibi alt panellere ayrılarak yukarda açıklanan işlemler her biri için tekrar edilmiştir. Alt paneller, ana panelin her bir kenarının orta noktaları birleştirilerek oluşturulmuştur. Bu alt panellerin ana panele olan etkileri hesaplanarak ana panelin kendi üzerindeki etkisi bulunmuştur.



Şekil Ek3.3 – Alt panellerin tanımlama sistemi

Ek 4 : Bilgisayar Programı

Aşağıda Gemi etrafındaki akımı çözmek için yazılan bilgisayar programının ana kısmı verilmiştir. Programlama dili olarak C++ kullanılmıştır.

```
//   Global variable tanimlar

      double numberofpanels;
      double numberofnodes;
      double hiz=10;

      #define n 2500

      double panel [n][4];
      double nodexyz [n][3];
      double panelcontrolpoint [n][3];
      double panelarea [n];
      double panelsubareas [n][4];
      double panelnormal [n][3];
      double panelsacp [n][12];  // panel subalanlarinin merkezleri icin
matrix acar


      double sigmamatrix [n][n];
      double cvektor [n];          // lineer denklem takımının esitlik
karsisindaki degerlerin vektoru sigmamatrix [n][n+1] anlamina gelir
      double sigmavektor [n]; // sigma kaynak siddetlerinin degerlerini
tasiyan vektor ,,,, lineer denklem sisteminin cozumunden cıkar
      double greenfunctions[n][n][3]; // green function'un dgdx, dgdy ve
dydz terimlerini depolar
      double panelhiz[n][3];  // Panel uzerindeki hizlar ( vx, vy, vz )
depolanir



/// Input ismi tanimlama ve hafizaya depolama

char filename;
char filetitle;

void CWt4Doc::OnFileOpen()
{
      CFileDialog dlg(TRUE, NULL, NULL, OFN_HIDEREADONLY |
OFN_OVERWRITEPROMPT,
                  "Flow Files (*.lev)|*.lev; |All Files (*.*)|*.*||", NULL
);


      if (dlg.DoModal() == IDOK)
      {

            SetCursor(AfxGetApp()->LoadStandardCursor(IDC_WAIT));

            CString filename = dlg.GetPathName();
            CString filetitle = dlg.GetFileTitle();
```

```
//----------------------------INPUT BOLUMU---------------------------------
//------------------------------------------------------------------------
//--------Ansys Input Dosyasindan , panellerin node numaralarini , --------
//--------nodelarin koordinatlarini belirtilmis matrislere depolar.--------
//------------------------------------------------------------------------

      SetCursor(AfxGetApp()->LoadStandardCursor(IDC_WAIT));

//---Input dosyasini acar-----------------------------------------
     ifstream    ijklfile;
       ijklfile.open((LPCTSTR)filename,ios::in);
//------------------------------------------------------------------



//---Output dosyasini acar-----------------------------------------
       ofstream    ijklfile1;

       ijklfile1.open("output.out",ios::out);
//------------------------------------------------------------------

//ijklfile1<< filename ;


//----node sayisini okur------------------------------------------
       ijklfile>>numberofnodes;
//----panel sayisini okur------------------------------------------
     ijklfile>>numberofpanels;





//------------------------------------------------------------------------
//    panel [panel numarasi][i,j,k,l] isimli matriste, panellerin node
numaralarini depolar
//------------------------------------------------------------------------
//    double panel [5000][4];

      for (int counter1=1;counter1<=numberofpanels;counter1++)
       {
            for (int counter2=1;counter2<=4;counter2++)
            {
            ijklfile>>panel [counter1][counter2];
            }
       }

      cout<<"paneller okundu \n";
//kontrol icin output verir

      ijklfile1<<"---------panel node numaralari--------------------------
\n" ;
```

```
      for (int counter3=1;counter3<=numberofpanels;counter3++)
       {
            for (int counter4=1;counter4<=4;counter4++)
            {
            ijklfile1<<panel [counter3][counter4];
            ijklfile1<<" " ;
            }
            ijklfile1<<"\n" ;
      }



//--------------------------------------------------------------------------
//    nodexyz [node sayisi][x,y,z] isimli matriste, node koordinatlarini
depolar
//--------------------------------------------------------------------------
//    double nodexyz [5000][3];

      for (int counter5=1;counter5<=numberofnodes;counter5++)
       {
            for (int counter6=1;counter6<=3;counter6++)
            {
            ijklfile>>nodexyz [counter5][counter6];
            }
       }
      cout<<"node koordinatlari okundu \n";

//kontrol icin output verir

      ijklfile1<<"--------- node koordinatlari--------------------------
\n" ;

      for (int counter7=1;counter7<=numberofnodes;counter7++)
       {
            for (int counter8=1;counter8<=3;counter8++)
            {
            ijklfile1<<nodexyz [counter7][counter8];
            ijklfile1<<" " ;
            }
            ijklfile1<<"\n" ;
      }




//--------------------------------------------------------------------------
//----------------------------Input bolumu biter----------------------------
//--------------------------------------------------------------------------


      }
}








void CWt4Doc::OnCalculatePanelnormalsareas()
{
```

```
      SetCursor(AfxGetApp()->LoadStandardCursor(IDC_WAIT));

//---Output dosyasini acar------------------------------------------
       ofstream    ijklfile2;
     ijklfile2.open("panelout.out",ios::out);


//--------------PANEL ALANLARI & CONTROL POINT LER & NORMAL DOGRULTU BOLUMU
//-----------------------------------------------------------------------
//---Panellerin ve subpanellerin alanlarini ve control pointlerini hesaplar
//---- 4 subpanel esas paneli kenar ortalarindan 2 ye bolerek elde edilir--
//--------ve bu kenarlarin kesisiminden ise control pointler bulunur-------
//-----------------------------------------------------------------------
//------Panelleri secerek i,j noktalarindan v1 vektoru , j,k noktalarindan
//--------v2 vektorunu olusturur ve cross product lari hesaplayarak duzleme
//--------dik olan normalin dogrultu vektorunu hesaplar ve ----------------
//--------panelnormal [panel no][vx_unit,vy_unit,vz_unit] matrisine depolar
//-----------------------------------------------------------------------



//   void icinde Global variable tanimlar
       int i,j,k,l;
       double ix,iy,iz,jx,jy,jz,kx,ky,kz,lx,ly,lz;
       double ijx,ijy,ijz,jkx,jky,jkz,klx,kly,klz,lix,liy,liz;
       double a1,a2,area;
       double v1x,v1y,v1z,v2x,v2y,v2z;
       double vx,vy,vz, vx_unit,vy_unit,vz_unit;




      for (int counter15=1;counter15<=numberofpanels;counter15++)
      {

             i = ( int ) panel [counter15][1];
             j = ( int ) panel [counter15][2];
             k = ( int ) panel [counter15][3];
             l = ( int ) panel [counter15][4];
             ix = nodexyz [i][1];
             iy = nodexyz [i][2];
             iz = nodexyz [i][3];
             jx = nodexyz [j][1];
             jy = nodexyz [j][2];
             jz = nodexyz [j][3];
             kx = nodexyz [k][1];
             ky = nodexyz [k][2];
             kz = nodexyz [k][3];
             lx = nodexyz [l][1];
             ly = nodexyz [l][2];
             lz = nodexyz [l][3];





//--------Panel alanlarini hesaplayan bolum--------------------------------
```

```
            ijx= jx-ix;
            ijy= jy-iy;
            ijz= jz-iz;

            jkx= kx-jx;
            jky= ky-jy;
            jkz= kz-jz;

            klx= lx-kx;
            kly= ly-ky;
            klz= lz-kz;

            lix= ix-lx;
            liy= iy-ly;
            liz= iz-lz;


            a1=0.5*sqrt((((ijy*jkz)-(jky*ijz))*((ijy*jkz)-
(jky*ijz)))+(((ijz*jkx)-(jkz*ijx))*((ijz*jkx)-(jkz*ijx)))+(((ijx*jky)-
(jkx*ijy))*((ijx*jky)-(jkx*ijy))));
            a2=0.5*sqrt((((kly*liz)-(liy*klz))*((kly*liz)-
(liy*klz)))+(((klz*lix)-(liz*klx))*((klz*lix)-(liz*klx)))+(((klx*liy)-
(lix*kly))*((klx*liy)-(lix*kly))));
            area=a1+a2;

            panelarea [counter15]=area;
//---------------------------------------------------------------------------



//--------Panel control pointlerini hesaplayan bolum----------------------

            panelcontrolpoint [counter15][1]=((ix+jx+kx+lx)/4);
            panelcontrolpoint [counter15][2]=((iy+jy+ky+ly)/4);
            panelcontrolpoint [counter15][3]=((iz+jz+kz+lz)/4);
//--------------------------------------------------------------------------


//--------Panel Normallerini hesaplayan bolum-----------------------------

             v1x= jx-ix;
             v1y= jy-iy;
             v1z= jz-iz;
             v2x= kx-jx;
             v2y= ky-jy;
             v2z= kz-jz;
      //----Cross product-------------------------------------------------

             vx= (v1y*v2z)-(v1z*v2y);
             vy= (v1z*v2x)-(v1x*v2z);
             vz= (v1x*v2y)-(v1y*v2x);

             vx_unit=vx/sqrt(vx*vx+vy*vy+vz*vz);
             vy_unit=vy/sqrt(vx*vx+vy*vy+vz*vz);
             vz_unit=vz/sqrt(vx*vx+vy*vy+vz*vz);

      //-------panelnormal matrisine depolar----------------------------
             panelnormal [counter15][1]=-vx_unit;
             panelnormal [counter15][2]=-vy_unit;
             panelnormal [counter15][3]=-vz_unit;
//--------------------------------------------------------------------------
```

```
//----------------------------Sub Paneller için hesaplar-------------------
//-----Ana panelin kenalarini ikiye bolerek 4 adet sub panel olusturulur ve
//---------bu sub panellerin alanlari ve control pointleri hesaplanir-----
//-----------------------------------------------------------------------
//ara noktalar icin ornek; i ve j noktalarinin orta noktasi araix , y ve z
dir
//-----------------------------------------------------------------------

            //----ara noktalar bulunur
      double
araix,araiy,araiz,arajx,arajy,arajz,arakx,araky,arakz,aralx,araly,aralz;


            araix=(ix+jx)/2;
            araiy=(iy+jy)/2;
            araiz=(iz+jz)/2;

            arajx=(jx+kx)/2;
            arajy=(jy+ky)/2;
            arajz=(jz+kz)/2;

            arakx=(kx+lx)/2;
            araky=(ky+ly)/2;
            arakz=(kz+lz)/2;

            aralx=(lx+ix)/2;
            araly=(ly+iy)/2;
            aralz=(lz+iz)/2;
            //-------------------------

            //---4 sub paneller alanlari ai,aj,ak ve al olarak tanimlanir--
            //---control pointleri ise aicp, ajcp, akcp ve alcp dir--------

            //---ai alani icin hesaplar------------------------------------
            //---i, arai, control point ve aral noktalari tarafindan
tanimlanir

            double
ai_ijx,ai_ijy,ai_ijz,ai_jkx,ai_jky,ai_jkz,ai_klx,ai_kly,ai_klz,ai_lix,ai_li
y,ai_liz;
            double ai_a1,ai_a2,ai_area;

            ai_ijx= araix-ix;
            ai_ijy= araiy-iy;
            ai_ijz= araiz-iz;

            ai_jkx= panelcontrolpoint [counter15][1]-araix;
            ai_jky= panelcontrolpoint [counter15][2]-araiy;
            ai_jkz= panelcontrolpoint [counter15][3]-araiz;

            ai_klx= aralx-panelcontrolpoint [counter15][1];
            ai_kly= araly-panelcontrolpoint [counter15][2];
            ai_klz= aralz-panelcontrolpoint [counter15][3];

            ai_lix= ix-aralx;
            ai_liy= iy-araly;
            ai_liz= iz-aralz;


            ai_a1=0.5*sqrt((((ai_ijy*ai_jkz)-
(ai_jky*ai_ijz))*((ai_ijy*ai_jkz)-(ai_jky*ai_ijz)))+(((ai_ijz*ai_jkx)-
```

```
(ai_jkz*ai_ijx))*((ai_ijz*ai_jkx)-(ai_jkz*ai_ijx)))+(((ai_ijx*ai_jky)-
(ai_jkx*ai_ijy))*((ai_ijx*ai_jky)-(ai_jkx*ai_ijy))));
            ai_a2=0.5*sqrt((((ai_kly*ai_liz)-
(ai_liy*ai_klz))*((ai_kly*ai_liz)-(ai_liy*ai_klz)))+(((ai_klz*ai_lix)-
(ai_liz*ai_klx))*((ai_klz*ai_lix)-(ai_liz*ai_klx)))+(((ai_klx*ai_liy)-
(ai_lix*ai_kly))*((ai_klx*ai_liy)-(ai_lix*ai_kly))));
            ai_area=ai_a1+ai_a2;

            panelsubareas [counter15][1]=ai_area;
            //---------------------------------------------------------------



            //---aj alani icin hesaplar-------------------------------------
            //---arai, j, araj ve control point noktalari tarafindan
tanimlanir

            double
aj_ijx,aj_ijy,aj_ijz,aj_jkx,aj_jky,aj_jkz,aj_klx,aj_kly,aj_klz,aj_lix,aj_li
y,aj_liz;
            double aj_a1,aj_a2,aj_area;


            aj_ijx= jx-araix;
            aj_ijy= jy-araiy;
            aj_ijz= jz-araiz;

            aj_jkx= arajx-jx;
            aj_jky= arajy-jy;
            aj_jkz= arajz-jz;

            aj_klx= panelcontrolpoint [counter15][1]-arajx;
            aj_kly= panelcontrolpoint [counter15][2]-arajy;
            aj_klz= panelcontrolpoint [counter15][3]-arajz;

            aj_lix= araix-panelcontrolpoint [counter15][1];
            aj_liy= araiy-panelcontrolpoint [counter15][2];
            aj_liz= araiz-panelcontrolpoint [counter15][3];


            aj_a1=0.5*sqrt((((aj_ijy*aj_jkz)-
(aj_jky*aj_ijz))*((aj_ijy*aj_jkz)-(aj_jky*aj_ijz)))+(((aj_ijz*aj_jkx)-
(aj_jkz*aj_ijx))*((aj_ijz*aj_jkx)-(aj_jkz*aj_ijx)))+(((aj_ijx*aj_jky)-
(aj_jkx*aj_ijy))*((aj_ijx*aj_jky)-(aj_jkx*aj_ijy))));
            aj_a2=0.5*sqrt((((aj_kly*aj_liz)-
(aj_liy*aj_klz))*((aj_kly*aj_liz)-(aj_liy*aj_klz)))+(((aj_klz*aj_lix)-
(aj_liz*aj_klx))*((aj_klz*aj_lix)-(aj_liz*aj_klx)))+(((aj_klx*aj_liy)-
(aj_lix*aj_kly))*((aj_klx*aj_liy)-(aj_lix*aj_kly))));
            aj_area=aj_a1+aj_a2;


            panelsubareas [counter15][2]=aj_area;
            //-----------------------------------------------------------


            //---ak alani icin hesaplar-------------------------------------
            //---control point, araj, k ve arak noktalari tarafindan
tanimlanir

            double
ak_ijx,ak_ijy,ak_ijz,ak_jkx,ak_jky,ak_jkz,ak_klx,ak_kly,ak_klz,ak_lix,ak_li
y,ak_liz;
```

```
            double ak_a1,ak_a2,ak_area;


            ak_ijx= arajx-panelcontrolpoint [counter15][1];
            ak_ijy= arajy-panelcontrolpoint [counter15][2];
            ak_ijz= arajz-panelcontrolpoint [counter15][3];

            ak_jkx= kx-arajx;
            ak_jky= ky-arajy;
            ak_jkz= kz-arajz;

            ak_klx= arakx-kx;
            ak_kly= araky-ky;
            ak_klz= arakz-kz;

            ak_lix= panelcontrolpoint [counter15][1]-arakx;
            ak_liy= panelcontrolpoint [counter15][2]-araky;
            ak_liz= panelcontrolpoint [counter15][3]-arakz;


            ak_a1=0.5*sqrt((((ak_ijy*ak_jkz)-
(ak_jky*ak_ijz))*((ak_ijy*ak_jkz)-(ak_jky*ak_ijz)))+(((ak_ijz*ak_jkx)-
(ak_jkz*ak_ijx))*((ak_ijz*ak_jkx)-(ak_jkz*ak_ijx)))+(((ak_ijx*ak_jky)-
(ak_jkx*ak_ijy))*((ak_ijx*ak_jky)-(ak_jkx*ak_ijy))));
            ak_a2=0.5*sqrt((((ak_kly*ak_liz)-
(ak_liy*ak_klz))*((ak_kly*ak_liz)-(ak_liy*ak_klz)))+(((ak_klz*ak_lix)-
(ak_liz*ak_klx))*((ak_klz*ak_lix)-(ak_liz*ak_klx)))+(((ak_klx*ak_liy)-
(ak_lix*ak_kly))*((ak_klx*ak_liy)-(ak_lix*ak_kly))));
            ak_area=ak_a1+ak_a2;


            panelsubareas [counter15][3]=ak_area;
            //---------------------------------------------------------------


            //---al alani icin hesaplar------------------------------------
            //---aral, control point, arak ve l noktalari tarafindan
tanimlanir

            double
al_ijx,al_ijy,al_ijz,al_jkx,al_jky,al_jkz,al_klx,al_kly,al_klz,al_lix,al_li
y,al_liz;
            double al_a1,al_a2,al_area;


            al_ijx= panelcontrolpoint [counter15][1]-aralx;
            al_ijy= panelcontrolpoint [counter15][2]-araly;
            al_ijz= panelcontrolpoint [counter15][3]-aralz;

            al_jkx= arakx-panelcontrolpoint [counter15][1];
            al_jky= araky-panelcontrolpoint [counter15][2];
            al_jkz= arakz-panelcontrolpoint [counter15][3];

            al_klx= lx-arakx;
            al_kly= ly-araky;
            al_klz= lz-arakz;

            al_lix= aralx-lx;
            al_liy= araly-ly;
            al_liz= aralz-lz;
```

```
            al_a1=0.5*sqrt((((al_ijy*al_jkz)-
(al_jky*al_ijz))*((al_ijy*al_jkz)-(al_jky*al_ijz)))+(((al_ijz*al_jkx)-
(al_jkz*al_ijx))*((al_ijz*al_jkx)-(al_jkz*al_ijx)))+(((al_ijx*al_jky)-
(al_jkx*al_ijy))*((al_ijx*al_jky)-(al_jkx*al_ijy))));
            al_a2=0.5*sqrt((((al_kly*al_liz)-
(al_liy*al_klz))*((al_kly*al_liz)-(al_liy*al_klz)))+(((al_klz*al_lix)-
(al_liz*al_klx))*((al_klz*al_lix)-(al_liz*al_klx)))+(((al_klx*al_liy)-
(al_lix*al_kly))*((al_klx*al_liy)-(al_lix*al_kly))));
            al_area=al_a1+al_a2;


            panelsubareas [counter15][4]=al_area;
            //---------------------------------------------------------------

        //------Panel Subalanlari Hesaplama Bolumu Biter------------------




//------Panel subalanlarin controlpointlerinin hesaplanmasi-------------
//------4 sub alanin merkezlerini --------------------------------------
//------panelsacp [panel
no][aix,aiy,aiz,ajx,ajy,ajz,akx,aky,akz,alx,aly,alz] matrisine depolar--

//------ai sub alani icin control point hesabi--------------------------

            panelsacp [counter15][1]=((ix+araix+panelcontrolpoint
[counter15][1]+aralx)/4);
            panelsacp [counter15][2]=((iy+araiy+panelcontrolpoint
[counter15][2]+araly)/4);
            panelsacp [counter15][3]=((iz+araiz+panelcontrolpoint
[counter15][3]+aralz)/4);

//------aj sub alani icin control point hesabi--------------------------

            panelsacp [counter15][4]=((araix+jx+arajx+panelcontrolpoint
[counter15][1])/4);
            panelsacp [counter15][5]=((araiy+jy+arajy+panelcontrolpoint
[counter15][2])/4);
            panelsacp [counter15][6]=((araiz+jz+arajz+panelcontrolpoint
[counter15][3])/4);


//------ak sub alani icin control point hesabi--------------------------

            panelsacp [counter15][7]=((panelcontrolpoint
[counter15][1]+arajx+kx+arakx)/4);
            panelsacp [counter15][8]=((panelcontrolpoint
[counter15][2]+arajy+ky+araky)/4);
            panelsacp [counter15][9]=((panelcontrolpoint
[counter15][3]+arajz+kz+arakz)/4);


//------al sub alani icin control point hesabi--------------------------

            panelsacp [counter15][10]=((aralx+panelcontrolpoint
[counter15][1]+arakx+lx)/4);
            panelsacp [counter15][11]=((araly+panelcontrolpoint
[counter15][2]+araky+ly)/4);
            panelsacp [counter15][12]=((aralz+panelcontrolpoint
[counter15][3]+arakz+lz)/4);
```

```
      }




      ijklfile2<<"----------panel normalleri nx  ny  nz-------------------
------ \n" ;

for (int counter16=1;counter16<=numberofpanels;counter16++)
       {
            for (int counter17=1;counter17<=3;counter17++)
            {
            ijklfile2<<panelnormal [counter16][counter17];
            ijklfile2<<" " ;
            }
            ijklfile2<<"\n" ;
      }
      ijklfile2<<"------------------------------------ \n" ;


      ijklfile2<<"---------panel alanlari--------------------------- \n" ;

      for (int counter100=1;counter100<=numberofpanels;counter100++)
       {
            ijklfile2<<counter100;
            ijklfile2<<" " ;
            ijklfile2<< panelarea [counter100];

            ijklfile2<<"\n" ;
      }

      ijklfile2<<"---------panel control pointler-------------------------
- \n" ;

      for (int counter101=1;counter101<=numberofpanels;counter101++)
       {
            for (int counter102=1;counter102<=3;counter102++)
            {
            ijklfile2<<panelcontrolpoint [counter101][counter102];
            ijklfile2<<" " ;
            }
            ijklfile2<<"\n" ;
      }

      ijklfile2<<"---------panel subareas--------------------------- \n" ;

      for (int counter103=1;counter103<=numberofpanels;counter103++)
       {
            ijklfile2<<counter103;
            ijklfile2<<" " ;
            for (int counter104=1;counter104<=4;counter104++)
            {

            ijklfile2<<panelsubareas [counter103][counter104];
            ijklfile2<<" " ;
            }
            ijklfile2<<"\n" ;
      }
```

```
      ijklfile2<<"---------panel subarea  control pointleri----------------
----------- \n" ;

      for (int counter105=1;counter105<=numberofpanels;counter105++)
       {
            ijklfile2<<counter105;
            ijklfile2<<" " ;
            for (int counter106=1;counter106<=12;counter106++)
            {

            ijklfile2<<panelsacp [counter105][counter106];
            ijklfile2<<" " ;
            }

            ijklfile2<<"\n" ;
      }


//-----------------------------------------------------------------------
-----------------
//----------------------------Normal dogrultu bolumu biter-----------------
-----------------------
//-----------------------------------------------------------------------
-----------------


ijklfile2<<"islem tamam \n" ;



}



void CWt4Doc::OnCalculateSigmamatrix()
{



//-----------------------------------SIGMA MATRIX OLUSTURULMASI-------------
//-----------------------------------------------------------------------
```

```
//------- sigma kaynak siddetlerinin hesabi icin lineer denklem sisteminin
olusturulmasi-----
//-- sigmamatrix [panelnumber][panelnumber]  matrixine terimleri depolar---
//-----------------------------------------------------------------------

      // Matrix için hafizada yer ayirilmasi
      // double sigmamatrix [5000][5000];
      // --------counter 18 hesap yapilan panel counter 19 ise 18 e etki
eden panellerdir---------


      SetCursor(AfxGetApp()->LoadStandardCursor(IDC_WAIT));


      //----------Rankine terim hesabi icin gerekli degiskenler------------
-----------------------
                        double dgdx,dgdy,dgdz;
                        double r,r1;
                        double coorx,coory,coorz;
                        double coore,coorn,coorf;
                        double nx;        // p (etki edilen) elemaninin
normal vektor bileseni x
                        double ny;        // p (etki edilen) elemaninin
normal vektor bileseni y
                        double nz;        // p (etki edilen) elemaninin
normal vektor bileseni z
                        double sq;        // q (etki eden) elemaninin alani

      //------------------------------------------------------------------
------------------------


      //----------Panelin kendi uzerindeki etkisini hesaplamak icic gerekli
degiskenler------------

                        double coor_p_x,coor_p_y,coor_p_z;
                        double
coor_ai_e,coor_ai_n,coor_ai_f,coor_aj_e,coor_aj_n,coor_aj_f,coor_ak_e,coor_
ak_n,coor_ak_f,coor_al_e,coor_al_n,coor_al_f;
                        double r_ai,r1_ai,r_aj,r1_aj,r_ak,r1_ak,r_al,r1_al;
                        double
dgdx_ai,dgdx_aj,dgdx_ak,dgdx_al,dgdy_ai,dgdy_aj,dgdy_ak,dgdy_al,dgdz_ai,dgd
z_aj,dgdz_ak,dgdz_al;
                        double s_ai,s_aj,s_ak,s_al;
                        double nx_p,ny_p,nz_p;  // Panelin normal vektor
bilesenleri
      //------------------------------------------------------------------
------------------------
for (int counter18=1;counter18<=numberofpanels;counter18++)
      {

            for (int counter19=1;counter19<=numberofpanels;counter19++)
            {

                  if (counter18 == counter19)
                        {
                        //----------------------Panellerin kendi
uzerlerindeki etkileri hesaplanir-----------------------
                        //-----------------Panele ait 4 sub alanin control
pointlerinin panelin control pointindeki--------
                        //-----------------etkisi bulunup toplanir ve
rankine terimlerindeki tekillik ortadan kalkar-------
```

```
                    //-----------------Panel merkezi x,y ve z dir ---
Subalanlarin merkezi e, n ve f dir----------------
                    //-------------------------------------------------
--------------------------------------------------------

                    //--------Panelin Control pointini alir-----------
----------------------------
                    coor_p_x = panelcontrolpoint [counter18][1];
                    coor_p_y = panelcontrolpoint [counter18][2];
                    coor_p_z = panelcontrolpoint [counter18][3];
                    //--------Panele ait ai sub panelinin control
pointini alir----------------------------
                    coor_ai_e = panelsacp [counter19][1];
                    coor_ai_n = panelsacp [counter19][2];
                    coor_ai_f = panelsacp [counter19][3];
                    //--------Panele ait aj sub panelinin control
pointini alir----------------------------
                    coor_aj_e = panelsacp [counter19][4];
                    coor_aj_n = panelsacp [counter19][5];
                    coor_aj_f = panelsacp [counter19][6];
                    //--------Panele ait ak sub panelinin control
pointini alir----------------------------
                    coor_ak_e = panelsacp [counter19][7];
                    coor_ak_n = panelsacp [counter19][8];
                    coor_ak_f = panelsacp [counter19][9];
                    //--------Panele ait al sub panelinin control
pointini alir----------------------------
                    coor_al_e = panelsacp [counter19][10];
                    coor_al_n = panelsacp [counter19][11];
                    coor_al_f = panelsacp [counter19][12];

                    //--------ai sub alaninin panele etkisi-----------
------------------------------------


                    r_ai = sqrt((coor_p_x-coor_ai_e)*(coor_p_x-
coor_ai_e)+(coor_p_y-coor_ai_n)*(coor_p_y-coor_ai_n)+(coor_p_z-
coor_ai_f)*(coor_p_z-coor_ai_f));
                    r1_ai = sqrt((coor_p_x-coor_ai_e)*(coor_p_x-
coor_ai_e)+(coor_p_y-coor_ai_n)*(coor_p_y-
coor_ai_n)+(coor_p_z+coor_ai_f)*(coor_p_z+coor_ai_f));

                    dgdx_ai= (((coor_p_x-coor_ai_e)/(r_ai*r_ai*r_ai))-
((coor_p_x-coor_ai_e)/(r1_ai*r1_ai*r1_ai)));
                    dgdy_ai= (((coor_p_y-coor_ai_n)/(r_ai*r_ai*r_ai))-
((coor_p_y-coor_ai_n)/(r1_ai*r1_ai*r1_ai)));
                    dgdz_ai= (((coor_p_z-coor_ai_f)/(r_ai*r_ai*r_ai))-
((coor_p_z+coor_ai_f)/(r1_ai*r1_ai*r1_ai)));

//------------------------------------------------------------------
---------------


//--------aj sub alaninin panele etkisi-----------------------------
------------

                    r_aj = sqrt((coor_p_x-coor_aj_e)*(coor_p_x-
coor_aj_e)+(coor_p_y-coor_aj_n)*(coor_p_y-coor_aj_n)+(coor_p_z-
coor_aj_f)*(coor_p_z-coor_aj_f));
                    r1_aj = sqrt((coor_p_x-coor_aj_e)*(coor_p_x-
coor_aj_e)+(coor_p_y-coor_aj_n)*(coor_p_y-
coor_aj_n)+(coor_p_z+coor_aj_f)*(coor_p_z+coor_aj_f));
```

```
                         dgdx_aj= (((coor_p_x-coor_aj_e)/(r_aj*r_aj*r_aj))-
((coor_p_x-coor_aj_e)/(r1_aj*r1_aj*r1_aj)));
                         dgdy_aj= (((coor_p_y-coor_aj_n)/(r_aj*r_aj*r_aj))-
((coor_p_y-coor_aj_n)/(r1_aj*r1_aj*r1_aj)));
                         dgdz_aj= (((coor_p_z-coor_aj_f)/(r_aj*r_aj*r_aj))-
((coor_p_z+coor_aj_f)/(r1_aj*r1_aj*r1_aj)));

//-----------------------------------------------------------------------
---------------


//--------ak sub alaninin panele etkisi-----------------------------------
------------

                         r_ak = sqrt((coor_p_x-coor_ak_e)*(coor_p_x-
coor_ak_e)+(coor_p_y-coor_ak_n)*(coor_p_y-coor_ak_n)+(coor_p_z-
coor_ak_f)*(coor_p_z-coor_ak_f));
                         r1_ak = sqrt((coor_p_x-coor_ak_e)*(coor_p_x-
coor_ak_e)+(coor_p_y-coor_ak_n)*(coor_p_y-
coor_ak_n)+(coor_p_z+coor_ak_f)*(coor_p_z+coor_ak_f));

                         dgdx_ak= (((coor_p_x-coor_ak_e)/(r_ak*r_ak*r_ak))-
((coor_p_x-coor_ak_e)/(r1_ak*r1_ak*r1_ak)));
                         dgdy_ak= (((coor_p_y-coor_ak_n)/(r_ak*r_ak*r_ak))-
((coor_p_y-coor_ak_n)/(r1_ak*r1_ak*r1_ak)));
                         dgdz_ak= (((coor_p_z-coor_ak_f)/(r_ak*r_ak*r_ak))-
((coor_p_z+coor_ak_f)/(r1_ak*r1_ak*r1_ak)));

//-----------------------------------------------------------------------
---------------

//--------al sub alaninin panele etkisi-----------------------------------
------------

                         r_al = sqrt((coor_p_x-coor_al_e)*(coor_p_x-
coor_al_e)+(coor_p_y-coor_al_n)*(coor_p_y-coor_al_n)+(coor_p_z-
coor_al_f)*(coor_p_z-coor_al_f));
                         r1_al = sqrt((coor_p_x-coor_al_e)*(coor_p_x-
coor_al_e)+(coor_p_y-coor_al_n)*(coor_p_y-
coor_al_n)+(coor_p_z+coor_al_f)*(coor_p_z+coor_al_f));

                         dgdx_al= (((coor_p_x-coor_al_e)/(r_al*r_al*r_al))-
((coor_p_x-coor_al_e)/(r1_al*r1_al*r1_al)));
                         dgdy_al= (((coor_p_y-coor_al_n)/(r_al*r_al*r_al))-
((coor_p_y-coor_al_n)/(r1_al*r1_al*r1_al)));
                         dgdz_al= (((coor_p_z-coor_al_f)/(r_al*r_al*r_al))-
((coor_p_z+coor_al_f)/(r1_al*r1_al*r1_al)));

//-----------------------------------------------------------------------
---------------


//------toplam etkiyi bulmak icin subalanlarin alanlarinin cagirilmasi-----
---------------

                         s_ai= panelsubareas [counter19][1];
                         s_aj= panelsubareas [counter19][2];
                         s_ak= panelsubareas [counter19][3];
                         s_al= panelsubareas [counter19][4];

//-----------------------------------------------------------------------
---------------
```

```
//------Panelin Normal vektor bilesenlerinin cagirilmasi-------------------
---------------

                        nx_p = panelnormal [counter18][1];
                        ny_p = panelnormal [counter18][2];
                        nz_p = panelnormal [counter18][3];

//-------------------------------------------------------------------------
---------------



                        sigmamatrix [counter18][counter19]=(-
2*3.141592654)+(((((dgdx_ai*s_ai)+(dgdx_aj*s_aj)+(dgdx_ak*s_ak)+(dgdx_al*s_
al))*nx_p)+(((dgdy_ai*s_ai)+(dgdy_aj*s_aj)+(dgdy_ak*s_ak)+(dgdy_al*s_al))*n
y_p)+(((dgdz_ai*s_ai)+(dgdz_aj*s_aj)+(dgdz_ak*s_ak)+(dgdz_al*s_al))*nz_p)))
;
                        //sigmamatrix
[counter18][counter19]=(1/(4*3.141592654))*(((((dgdx_ai*s_ai)+(dgdx_aj*s_aj
)+(dgdx_ak*s_ak)+(dgdx_al*s_al))*nx_p)+(((dgdy_ai*s_ai)+(dgdy_aj*s_aj)+(dgd
y_ak*s_ak)+(dgdy_al*s_al))*ny_p)+(((dgdz_ai*s_ai)+(dgdz_aj*s_aj)+(dgdz_ak*s
_ak)+(dgdz_al*s_al))*nz_p)));


                        greenfunctions[counter18][counter19][1]=-
((dgdx_ai*s_ai)+(dgdx_aj*s_aj)+(dgdx_ak*s_ak)+(dgdx_al*s_al));
                        greenfunctions[counter18][counter19][2]=-
((dgdy_ai*s_ai)+(dgdy_aj*s_aj)+(dgdy_ak*s_ak)+(dgdy_al*s_al));
                        greenfunctions[counter18][counter19][3]=-
((dgdz_ai*s_ai)+(dgdz_aj*s_aj)+(dgdz_ak*s_ak)+(dgdz_al*s_al));

                        }

                  if (counter18 != counter19)
                        {

//-------------Etki eden panellerin hesap yapilan panele etkisi hesaplanir
//------------------------------------------------------------------------
//--------------Green Fonksiyonunun Rankine Terimlerinin Hesaplanmasi
//------------------------------------------------------------------------
//---------------[dg/dx*nx(p counter18)+dg/dy*ny(p counter18)+dg/dz*nz(p
counter18)]*DeltaS(Q counter19)
//-----------------------------------------------------------------------
//-----Rankine terimlerinin x, y ve z 'e gore turevlerinin hesabi---------


// burada r = [(x-e)2+(y-n)2+(z-f)2]^^1/2-----------------r1 = [(x-e)2+(y-
n)2+(z+f)2]^^1/2
// x,y ve z hesap yapilan alanin merkezinin koordinatlari,
// e,n ve f etki eden alanlarin merkez koordinatlaridir

                        // alan merkez kooridanatlarinin okunmasi
                        coorx = panelcontrolpoint [counter18][1];
                        coory = panelcontrolpoint [counter18][2];
                        coorz = panelcontrolpoint [counter18][3];

                        coore = panelcontrolpoint [counter19][1];
                        coorn = panelcontrolpoint [counter19][2];
                        coorf = panelcontrolpoint [counter19][3];
```

```
                        r = sqrt((coorx-coore)*(coorx-coore)+(coory-
coorn)*(coory-coorn)+(coorz-coorf)*(coorz-coorf));
                        r1 = sqrt((coorx-coore)*(coorx-coore)+(coory-
coorn)*(coory-coorn)+(coorz+coorf)*(coorz+coorf));


                        dgdx= -(((coorx-coore)/(r*r*r))-((coorx-
coore)/(r1*r1*r1)));
                        dgdy= -(((coory-coorn)/(r*r*r))-((coory-
coorn)/(r1*r1*r1)));
                        dgdz= -(((coorz-coorf)/(r*r*r))-
((coorz+coorf)/(r1*r1*r1)));

                        greenfunctions[counter18][counter19][1]=dgdx;
                        greenfunctions[counter18][counter19][2]=dgdy;
                        greenfunctions[counter18][counter19][3]=dgdz;

                        nx = panelnormal [counter18][1];
                        ny = panelnormal [counter18][2];
                        nz = panelnormal [counter18][3];

                        sq = panelarea [counter19];

                        sigmamatrix
[counter18][counter19]=/*(1/(4*3.141592654))**/(((dgdx*nx)+(dgdy*ny)+(dgdz*
nz))*sq);




                        }

            }






      }

      //------------------------------------------------------------------
-------------------------




      //-------------kontrol icin sigmamatrix [][] 'in outputa
yazdirilmasi---------------
      //------------------------------------------------------------------
--------------
      //---Output dosyasini acar----------------------------------------
       ofstream    ijklfile3;
     ijklfile3.open("sigmamatrixout.out",ios::out);
      //-----------------------------------------------------------------


for (int counter20=1;counter20<=numberofpanels;counter20++)
      {

            for (int counter21=1;counter21<=numberofpanels;counter21++)
            {
```

```
                        ijklfile3<<sigmamatrix [counter20][counter21];
                        ijklfile3<<" " ;

            }
                        ijklfile3<<"\n" ;
      }
      //----------------------- kontrol yazdirmasi biter------------------
----------------


}



//------------------------------------------------------------------------
--------------
//---------------------------------SOLUTION OF MATRIX---------------------
--------------
//------------------------------------------------------------------------
--------------


void CWt4Doc::OnSolutionSolvefordoublebody()
{
      SetCursor(AfxGetApp()->LoadStandardCursor(IDC_WAIT));

//---Output dosyasini acar------------------------------------------
       ofstream     ijklfile4;
       ijklfile4.open("solve.out",ios::out);
//-----------------------------------------------------------------



//-------cvektor[]'un yani sigmamatrix[][] 'in numberofpanels+1 inci
terimlerinin model hizina bagli olarak bulunmasi-----
//-------burada denklemler U akis hizi * panelin nx dogrultu normali ile
carpimi ile esitlenir-----------




            for (int counter26=1;counter26<=numberofpanels;counter26++)
            {
                  double nx = panelnormal [counter26][1];
            //    double hiz = 10;
                  cvektor[counter26]=nx*hiz;

            }

for (int counter101=1;counter101<=numberofpanels;counter101++)
      {


                        ijklfile4<<cvektor[counter101];
                        ijklfile4<<" " ;

                        ijklfile4<<"\n" ;
      }
```

```
//-------Naive Gauss Elimination-------------------------------------
//-------------------------------------------------------------------
//-------Forward Elimination-----------------------------------------

for (int k=1;k<=(numberofpanels-1);k++)
      {

            for (int i=(k+1);i<=numberofpanels;i++)
            {
                  double factor =sigmamatrix[i][k]/sigmamatrix[k][k];

                  for (int j=(k+1);j<=numberofpanels;j++)
                  {
                        sigmamatrix[i][j]=sigmamatrix[i][j]-
factor*sigmamatrix[k][j];

                  }
                        cvektor[i]=cvektor[i]-factor*cvektor[k];
            }

      }


//--------Backward Subsition------------------------------------------
-


      sigmavektor[(int)
numberofpanels]=(cvektor[(int)numberofpanels]/sigmamatrix[(int)
numberofpanels][(int) numberofpanels]);


for (int i=((int) numberofpanels-1);i>=1;i--)
      {
            double sum=0;
            for (int j=(i+1);j<=numberofpanels;j++)
            {

            sum = sum+sigmamatrix[i][j]*sigmavektor[j];

            }

            sigmavektor[i]=(cvektor[i]-sum)/sigmamatrix[i][i];
      }
//------------------------------------------------------------------
----


//----------Bulunan kaynak degerleri output dosyasina yazdirilir---------
---
                        ijklfile4<<"kaynak degerleri\n" ;
for (int counter102=1;counter102<=numberofpanels;counter102++)
      {

                        ijklfile4<<sigmavektor[counter102];
                        ijklfile4<<" " ;

                        ijklfile4<<"\n" ;
```

```
      }
//----------------------------------------------------------------------------
-----

//----------Paneller uzerindeki Hizlari hesaplama bolumu ------------------
-----


for (int counter110=1;counter110<=numberofpanels;counter110++)
      {
            panelhiz[counter110][1]=0;
            panelhiz[counter110][2]=0;
            panelhiz[counter110][3]=0;

            for (int counter111=1;counter111<=numberofpanels;counter111++)
            {


      panelhiz[counter110][1]+=((sigmavektor[counter111]*greenfunctions[cou
nter110][counter111][1]*panelarea [counter111]));

      panelhiz[counter110][2]+=((sigmavektor[counter111]*greenfunctions[cou
nter110][counter111][2]*panelarea [counter111]));

      panelhiz[counter110][3]+=((sigmavektor[counter111]*greenfunctions[cou
nter110][counter111][3]*panelarea [counter111]));

            }
      }

//------Hizlarin yazdirilmasi-------------------------------------------
      ofstream     ijklfile20;
      ijklfile20.open("panelhiz.out",ios::out);
      for (int counter500=1;counter500<=numberofpanels;counter500++)
      {
            for (int counter501=1;counter501<=3;counter501++)
            {
                        ijklfile20<<panelhiz[counter500][counter501];
                        ijklfile20<<" " ;
            }
                        ijklfile20<<"\n" ;
      }
//----------------------------------------------------------------------

}
```

ÖZGEÇMİŞ

Doğum tarihi 26.12.1977

Doğum yeri	Bursa	
Lise	1988-1994	Bursa Erkek Lisesi
Lisans	1994-1999	Yıldız Üniversitesi Makine Fakültesi Gemi İnşaatı Mühendisliği Bölümü
Yüksek Lisans	1999-2002	Yıldız Üniversitesi Makine Fakültesi Gemi İnşaatı Mühendisliği Bölümü

Çalıştığı kurum(lar)

1999-Devam ediyor	Delta Denizcilik Mühendislik A.Ş