

KOR İÇİ YAKIT YÖNETİM KOD SİSTEMİ GELİŞTİRİLMESİ

**DEVELOPMENT OF IN CORE FUEL MANAGEMENT CODE
SYSTEM**

ERHAN ŞENLİK

Prof. Dr. MEHMET TOMBAKOĞLU

Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin
Nükleer Enerji Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2014

ERHAN ŐENLİK' in hazırladığı "**Kor İçi Yakıt Yönetim Kod Sistemi Geliştirilmesi**" adlı bu çalışma aŐağıdaki jüri tarafından **NÜKLEER ENERJİ MÜHENDİSLİĐİ ANABİLİM DALI'** nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Ayhan Yılmaz
BaŐkan

.....

Prof. Dr. Mehmet Tombakođlu
DanıŐman

.....

Prof. Dr. Süleyman Ali Tuncel
Üye

.....

Prof. Dr. Cemal Niyazi Sökmen
Üye

.....

Doç. Dr. Cemil Kocar
Üye

.....

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak onaylanmıştır.

Prof. Dr. Fatma SEVİN DÜZ
Fen Bilimleri Enstitüsü Müdürü

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- tez içindeki bütü bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

29/01/2014

Erhan ŞENLİK

ÖZET

KOR İÇİ YAKIT YÖNETİM KOD SİSTEMİ GELİŞTİRİLMESİ

Erhan Şenlik

Yüksek Lisans,Nükleer Enerji Mühendisliği Bölümü

Tez Danışmanı: Prof. Dr. MEHMET TOMBAKOĞLU

Ocak 2014, 55 sayfa

Bu çalışma bir boyutlu kor yakıt yükleme düzeni ve Almaraz II Nükleer Güç Santralinin yakıt yükleme düzeni esas alınarak, yakıt yükleme işlemindeki hesaplamaları anlamayı ve bunları kolaylaştırmayı hedefleyen altı farklı bilgisayar programından oluşmaktadır Üç program çözücü olarak yapılacak işlemler için hesaplamaları yapan ve hızlı sonuç vermesi istendiğinden FORTRAN programlama dilinde yazılmış, diğer üç program ise bu iki programa girdi dosyalarını oluşturmak için Python programlama dilinde yazılmış grafiksel kullanıcı arayüz programlarından oluşmaktadır.Hesaplama programları 1DNodal, RPM ve RPM-Genetik , grafiksel ara-yüz programları ise Py1DNodal, PyRPM ve PyRPM-Genetik tir. 1DNodal program paketi bir boyutlu kor yükleme düzeni esas alınarak hazırlanmıştır. Py1DNodal programı ise bu programa belirlenmiş yakıt tipleri arasından istenen yakıtların reactor koruna yerleştirilmesini sağlayan grafiksel kullanıcı arayüzü olarak tasarlanmıştır. RPM hesaplama programında yakıt düzeni,yanma oranları ve güç seviyeleri kullanıcı tarafından PyRPM programı kullanılarak tayin edilmesini sağlayacak şekilde tasarlanmıştır. RPM-Genetik hesaplama programında kullanıcı yakıt demetlerinin sayılarını ve genetik algoritma için gerekli parametreleri PyRPM-Genetik grafiksel ara-yüzü üzerinden girerek hesaplamayı yapacak şekilde tasarlanmıştır. Bütün programlar farklı girdiler kullanılarak test edilmiş ve son geliştirici tarafından gelişimine devam edebilecek şekilde sonlandırılmıştır.

Anahtar Kelimeler: Nükleer Yakıt Yönetimi, Genetik Algoritma

ABSTRACT

DEVELOPMENT OF IN CORE FUEL MANAGEMENT CODE SYSTEM

Erhan ŞENLİK

Master of Science, Department of Nuclear Engineering

Supervisor: Prof. Dr. Mehmet TOMBAKOĞLU

January 2014, 55 pages

This study consists of six different computer programs. These programs are based on one dimensional core loading pattern and fuel loading pattern_of Almaraz II Nuclear Power Plant for easy understanding of core loading calculations.

Three programs are solvers and they are coded in FORTRAN to acquire results quickly. (because of the language's fast solving capabilities) Remaining three programs are graphical user interface programs which were coded in Python programming language. Calculation programs are; 1DNodal, RPM and RPM-Genetic, graphical user interface programs are; Py1DNodal, PyRPM, PyRPM-Genetic.

1DNodal software is based on 1 dimensional core loading pattern. Py1DNodal software is a graphical user interface for loading pre-chosen fuel types as an input to the 1DNodal software.

PyRPM code is for determination of fuel pattern, burn-up and power levels for RPM calculation software. PyRPM-Genetik graphical interface is designed for supplying number of fuel assemblies and required input parameters of genetic algorithm to RPM-Genetik software. All software has been tested with different inputs and all are open for further development.

Keywords: nuclear fuel management, genetic algorithm

TEŞEKKÜR

Lisans eğitimim ile tanışma fırsatı yakaladığım ,her zaman engin bilgi ve tecrübesi ile yanımda olan, beni destekleyen, bu tez çalışmasını bitirmemi ve yapmamı sağlayan değerli Hocam, Rehberim Sayın Prof. Dr. Mehmet Tombakoğlu'na en içten duygularıyla teşekkürü bir borç bilirim.

Ayrıca benden desteklerini esirgemeyerek her zaman bilgi ve birikimleriyle bana yol gösteren ve tez değerlendirme komitemde olma nezaketini gösteren değerli hocalarım Sayın Prof. Dr. Cemal Niyazi Sökmen'e, Sayın Prof. Dr. Süleyman Ali Tuncel'e, Sayın Doç. Dr. Ayhan Yılmaz'ın, Sayın Doç. Dr. Cemil Kocar'a teşekkür ederim.

Tez çalışmam boyunca bilgisayar programlarım hakkındaki eleştiri, yardım ve desteği ile yetinmeyip her zaman yanımda olan değerli dostum Nükleer Enerji Yüksek Mühendisi Atilla Yıldız'a minnettarlığımı sunarım.

Çalışmam boyunca bana olan manevi desteklerinden dolayı Hacettepe Üniversitesi Nükleer Enerji Mühendisliği Ailesi'ne teşekkür ederim.

Son olarak her zaman yanımda olan, beni destekleyen, sevgileri ile beni daha ileriye taşıyan aileme teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER	viii
KISALTMALAR.....	ix
1. GİRİŞ	1
2. KOR YÜKLEME DÜZENİ HESAPLANAN NÜKLEER GÜÇ SANTRALİ:.....	3
2.1. Almaraz Nükleer Güç Santrali:	3
3. GELİŞTİRME ARAÇLARI	10
3.1. FORTRAN Programla Dili	10
3.2. Python Programlama Dili.....	11
3.3. Qt Kütüphanesi.....	12
4. MATEMATİKSEL MODELLER	13
4.1. Bir-buçuk Grup Nötron Difüzyon Denklemi Yaklaşımı.....	14
4.2. Levha Geometrisi için Güç Paylaşım Denkliklerinin Türetilmesi [11]	18
4.3. Kare Şeklindeki Düğüm Noktaları için Güç Paylaşım Denkliklerinin Türetilmesi [11].....	21
5. PROGRAMLAMADA GENETİK ALGORİTMA YAKLAŞIMI	23
6. PROBLEMLERİN TANIMLANMASI VE GRAFİKSEL KULLANICI ARAYÜZLERİ	24
6.1. Tek Boyutlu ve İki Boyutlu Kor Yükleme Hesaplaması için RPM Programı ve Grafiksel AraYüz(PyRPM).....	25
6.2. Kor Yükleme Düzeni için RPM-GENETİK ve Grafiksel Kullanıcı Arayüzü(PyRPM-GENETIC)	33
7. ÖRNEK PROBLEMLERİN UYGULANMASI.....	40
7.1. Tek Boyutlu Kor Yükleme Düzeni İçin Çeşitli Uygulamalar	40
7.2. Almaraz NGS İki Boyutlu Kor Yükleme Düzeni İçin Çeşitli Uygulamalar	44
7.3. Almaraz NGS Kor Yükleme Düzeni için Genetik Algoritma Uygulaması.....	49
8. SONUÇLAR.....	53

KAYNAKLAR.....	56
ÖZGEÇMİŞ	57

ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 2.1 Reaktör kor geometrisi [5]	7
Şekil 2.2 Almaraz NGS çubuk kümesi ve kontrol çubukları demeti [5].....	8
Şekil 2.3 Almaraz NGS yakıt demeti kesit görüntüsü 17 x 17 [5]	9
Şekil 4.1 Levha Geometrisi	18
Şekil 4.2 Kartezyen geometri için düğüm noktaları [12]	21
Şekil 6.1 Almaraz Kor Yükleme Düzeni [17].....	24
Şekil 6.2 1dNodal Programı arayüzü.....	25
Şekil 6.3 Çıktı terminali.....	26
Şekil 6.4 PyRPM program ana düzenleme sayfası	27
Şekil 6.5 PyRPM program ana çıktı sayfası	28
Şekil 6.6 Yakıt demetlerinin seçimi.....	28
Şekil 6.7 Yakıt demetlerinden tipi belirlenmiş olanların görüntüsü	29
Şekil 6.8 PyRPM program çıktı terminali.....	30
Şekil 6.9 Program ana döngü ve iterasyon sayılarının belirlenmesi.....	31
Şekil 6.10 PyRPM Program Döngü Değerlerinin Girilmesi Ekranı	32
Şekil 6.11 PyRPM-Genetik Ana Sayfası Görüntüsü.....	33
Şekil 6.12 Genetik Parametreler Giriş Penceresi	34
Şekil 6.13 Yakıt Demetleri Parametreleri Giriş Penceresi	36
Şekil 6.14 Yakıt Demetleri Sızıntı Yüzey ve Reaktivite Değerleri Giriş Penceresi	37
Şekil 6.15 Genetik Algoritma Akı Şeması [18].....	38
Şekil 7.1 Py1DNodal Giriş Arayüzü.....	40
Şekil 7.2 Py1DNodal Çıktı Terminali	41
Şekil 7.3 1DNodal 1.Yakıt Tipi için Normalize Güç Değerleri.....	42
Şekil 7.4 1DNodal 1.Yakıt Tipi için Yanma Oranları.....	42
Şekil 7.5 1DNodal 1. Ve 3. Tip Yakıtların Yanma Oranları.....	43

Şekil 7.6 PyRPM Giriş Penceresi	45
Şekil 7.7 PyRPM Giriş Penceresinde Yakıtların Düzenlenmiş Görüntüsü	45
Şekil 7.8 Normalize Güç Oranları	46
Şekil 7.9 Yakıt Demetleri Yanma Oranları.....	47
Şekil 7.10 Boron Konsantrasyonu Değişimi	47
Şekil 7.11 Yakıt Demetleri ve Popülasyon Sayısı Penceresi	49
Şekil 7.12 Genetik Parametreler Giriş Penceresi	50
Şekil 7.13 PyRPMGenetic Ana Pencere	51
Şekil 7.14 Çıktı Dosyası Sonucu	52
Şekil 8.1 PyRPM Program Çıktı Sayfası	54
Şekil 8.2 PyRPM-Genetic Program Çıktı Sayfası.....	54

ÇİZELGELER

	<u>Sayfa</u>
Çizelge 2.1 Almaraz 1 NGS teknik özellikleri [1,2,3,4]	3
Çizelge 2.2 Almaraz NGS test için verilen teknik özellikleri [5]	4
Çizelge 7.1 Almaraz NGS yakıt tipleri	44

KISALTMALAR

NPP:	Nuclear Power Plant (Nükleer Güç Santrali)
NGS:	Nükleer Güç Santrali
Almaraz:	Almaraz II Nükleer Güç Santrali
GKA:	Grafiksel Kullanıcı Ara-yüzü
GA:	Genetik Algoritma

1. GİRİŞ

Günümüzde teknolojinin gelişmesiyle birlikte artan enerji talebi pek çok farklı kaynaktan sağlanmaktadır. Bu kaynaklar arasında başlıca olanlar fosil yakıtlar, nükleer yakıtlar ve temiz tükenmez (güneş, su, rüzgar) gibi enerji kaynaklarıdır. Ülkelerin politik, stratejik görüşleri ve teknolojik gelişmişlikleri ölçüsünde bahsi geçen kaynaklar farklı yüzdelerle neredeyse tüm dünyada kullanılmaktadır.

II. Dünya savaşının ardından artan enerji talebi ve gelişmiş askeri endüstri ile birlikte ortaya nükleer enerjinin elektrik enerjisi üretiminde kullanımı söz konusu olmuştur. Bu konuda yapılan araştırmalarla birlikte nükleer enerji, elektrik enerjisi üretimi halkasına katılmıştır. Diğer enerji kaynaklarının aksine nükleer yakıtlardan enerji elde edilmesi daha çok araştırma, analiz ve karmaşık mühendislik hesaplamaları gerektirmektedir. Tüm bu çalışmalar dünyada nükleer yakıttan enerji elde edilmesi oranını zaman içerisinde daha yüksek oranlara ulaştırmış ve artmıştır. Bunun başlıca sebepleri ucuz, sürdürülebilir, çevreci ve yükselen talebi karşılayabilmesidir. Bu bağlamda dünyada en çok kullanılan nükleer güç santrali tipi basınçlı su reaktörü (PWR)' dir. Basınçlı su reaktörleri dizayn ve mühendislik açısından üretici firmaya göre farklılık gösterse de temelde aynı prensipler üzerine çalışmaktadır.

Nükleer reaktör korundan geçen soğutucuya Nükleer Enerjiden ortaya çıkan ısı enerjisinin aktarılmasıdır.

Nükleer Güç Santrallerinde fosil yakıt esaslı güç santrallerinin aksine yakıt yüklemeleri sürekli değil kesikli olarak yapılır. Yani belirli periyotlarla(zaman aralıklarıyla (12 ay/18 ay gibi) nükleer yakıtların konumları değiştirilir ve tükenen nükleer yakıtlar yenileriyle değiştirilir. Yakıt değişimlerinin süresi tasarımdan tasarıma değişiklik göstermektedir. Güç santrallerin de ekonomik olarak esas olan, aralıksız olarak elektrik üretimini sağlamak ve mümkün olduğunca sürekli enerji arzında bulunmaktır.

Yakıtların reaktörde kullanımı esnasında maksimum enerjiyi güvenli bir şekilde üretebilmek için farklı yakıtların reaktör korundaki konumlarını hesaplamak gerekir. Bu yükleme düzenleri nükleer reaktörün türüne, kullandığı yakıtın tipine ve yakıt çevrimin süresine bağlıdır.

Bu hesaplamaların karmaşıklığı sebebiyle analitik olarak herhangi bir bilgisayar veya hesaplama aracı desteği almadan yapılması pek mümkün değildir. Nükleer Güç Santrallerinin yapılmaya ve araştırılmasına başlandığı tarihten itibaren tüm bu karmaşık hesaplamalar bilgisayar programları yazılarak yapılmıştır.

Bu tez çalışmasında İspanya'da bulunan nükleer güç santrali (NGS) Almaraz NGS'nin yeniden yakıt yüklemesi modellemeleri için daha önceden Hacettepe Üniversitesi Nükleer Enerji Mühendisliği bölümü tarafından geliştirilmiş olunan RPM (Reloading Power Mapping) bilgisayar programının yeniden düzenlenerek bir grafiksel kullanıcı ara-yüzüne kavuşturulması sağlanmıştır. Bu arayüz ile son kullanıcı (end user) veya son geliştiricinin yakıt yükleme ve yakma stratejileri açısından konuyu daha iyi anlaması ve daha hızlı sonuca ulaşması hedeflenmiştir.

Bu amaca yönelik olarak bir boyutta levha şeklinde homojen bölgelerden oluşan reaktör modeli için bilgisayar programı grafiksel arayüzü ile birlikte hazırlanmıştır. PWR tipi reaktörlerden biri olan Almaraz Reaktörü içinde iki boyutlu hesap yapabilen RPM kodu geliştirilerek hesaplamalarda kullanılmıştır.

Hazırlanan grafiksel kullanıcı arayüzleri(GKA) üç ana FORTRAN programı üzerine oturtulmuştur. Her bir GKA farklı problemler için tasarlanmıştır. Bir boyutta ve iki boyutta model problemler çalışılmış ve Almaraz-NGS kor yükleme düzenleri içinde modeller oluşturulmuştur.

Bu çalışma; kor yükleme düzenleri hesaplanan nükleer güç santrallerinin teknik özellikleri ve nitelikleri, tanımlanan problemlerin bilgisayar esaslı çözümü için uyarlanan programlama dilleri ve kütüphaneleri, problemlerin çözümünde kullanılan matematiksel yöntemler ve bunların bilgisayar programlarına uyarlanması, geliştirilen kullanıcı arayüzlerinin geliştirilme aşamaları ve son olarak sonuçların değerlendirilmesi olarak beş temel bölümden oluşmaktadır.

2. KOR YÜKLEME DÜZENİ HESAPLANAN NÜKLEER GÜÇ SANTRALİ:

Hazırlanan bilgisayar programları başlangıç ve sonraki periyotlar için kor yükleme düzenlerini hesaplamayı amaçlamaktadır. Kor yükleme düzenleri için gerekli parametreler reaktörün özellikleri kullanılarak tasarlanmıştır. İki kullanıcı grafik arayüzü de Almaraz Nükleer Reaktörünün özellikleri baz alınarak tasarlanmıştır.

2.1. Almaraz Nükleer Güç Santrali:

Almaraz NGS: İspanya'nın Almaraz kentinde bulunan Amerikan Westinghouse firması tarafından tasarlanan ve yapılan nükleer santraldir [1]. Program çalışmasında bu reaktörün yakıt çubuklarının özellikleri ve kor düzeni esas alınmıştır. Almaraz NGS 'ne ait bazı veriler aşağıda çizelge halinde detaylı olarak verilmiştir.

Çizelge 2.1 Almaraz 1 NGS teknik özellikleri [1,2,3,4]

Veriler:	Birimi:	Değeri:
Reaktör Tipi ve Modeli		PWR(Westinghouse)
Nominal Elektrik Çıktısı	MW	973.5
Nominal Termal Güç	MW	2696
Soğutucu Tipi		Hafif Su
Soğutucu Hacmi	m ³	271
Birinci Devre Hacmi	m ³	287.5
Yavaşlatıcı Tipi		Hafif Su
Yakıt Tipi(oksit/metal)		UO ₂ Pelet
Tipik Yakıt Zenginliği	% kütle	Maks. 3.1 %
Yakıt Zarfı		Zircolay-4
Yakıt Elemanı Sayısı		157
Korteki U Kütlesi	Ton	82.195
Kontrol Çubuğu Tipi		Ag-In-Cd paslanmaz çelik zarf içerisinde

Kontrol Çubuğu Sayısı		48
İkincil Kapatma Sistemi		Borik Asit Enjeksiyonu
Basınç Kabı Materyali		Karbon-Çelik/Paslanmaz Çelik Zarf
Basınç Kabı Çapı	m	4
Basınç Kabı Yüksekliği	m	13
Basınç Kabı Et Kalınlığı	m	0.2
Soğutucu Basıncı	MPa	15.5
Reaktör Girişindeki Soğutucu Sıcaklığı	°C	292
Reaktör Çıkışındaki Soğutucu Sıcaklığı	°C	327
Birincil Devre Döngü Sayısı		3
Birincil Soğutucu Pompa Sayısı		3
Birincil Devre Soğutucu Akış hızı	t/s	13.08
Buhar Üretici Sayısı		3
Buhar Üretici Tipi		U-tube (Siemens 61W/D3)

Almaraz NGS' nin tüm verileri kullanılarak yeniden yakıt yükleme düzeni için test problemleri oluşturulmuş ve aşağıda verilmiş olan bilgiler doğrultusunda bilgisayar programlarının yazılması ve test edilmesi istenmiştir [5].

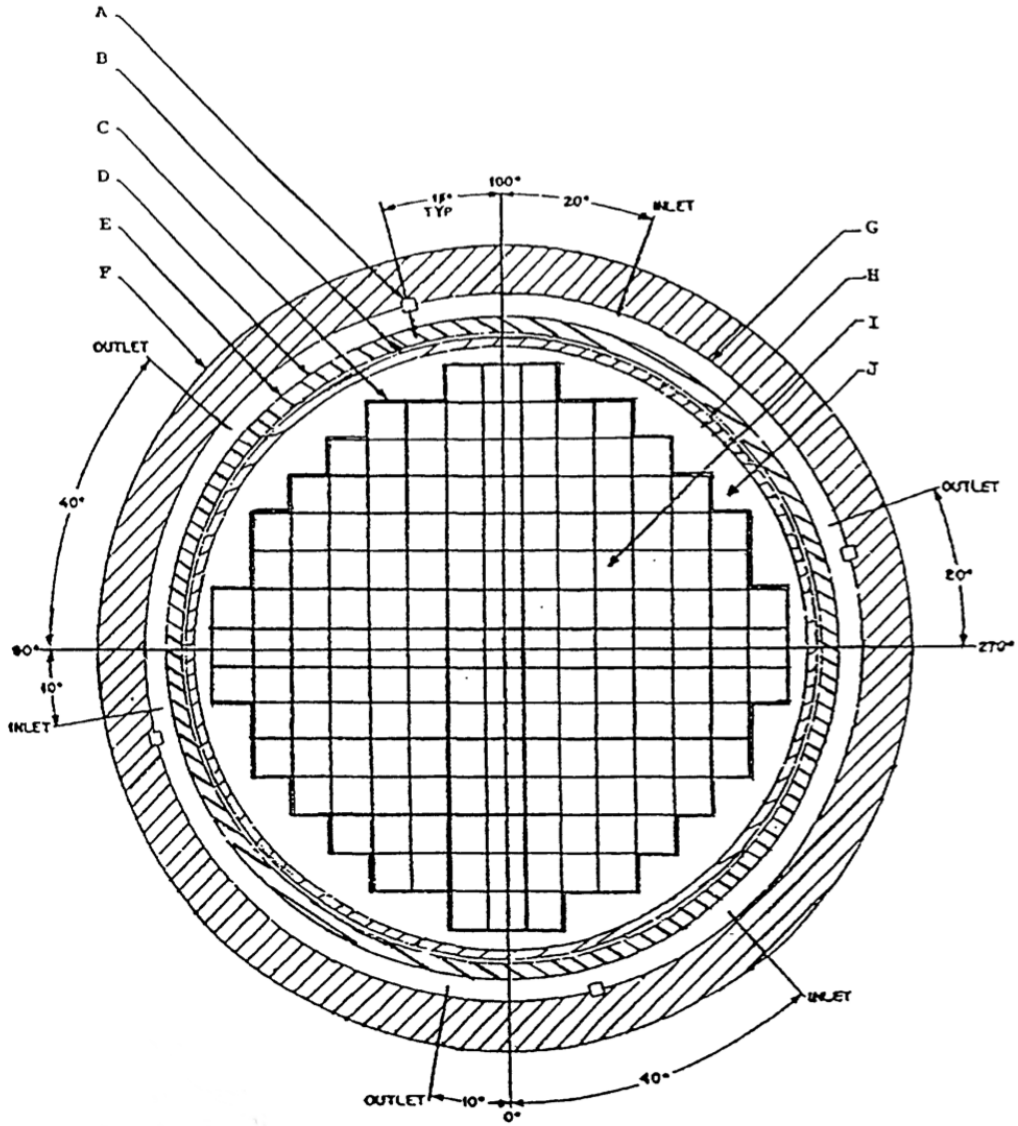
Çizelge 2.2 Almaraz NGS test için verilen teknik özellikleri [5]

Veriler:	Değeri:	Ref.:
Reaktör Özellikleri:		
Termal Güç	2686 MW	7
Döngü Sayısı	3	7
Yakıttan Üretilen Isı	97.4 %	6
Soğutucu Özellikleri:		
Nominal Basınç	155.137 bar	6
Korteki Akış hızı	1.38×10^4 kg/s	7
HFP Giriş Sıcaklığı	291.4	7

HFP Ortalama Kor Çıkış Sıcaklığı	326 °C	6	
HFP Ortalama Yavaşlatıcı Sıcaklığı	309.9 °C	6	
HFP Ortalama Yakıt Zarfı Sıcaklığı	340 °C	12	
HFP Ortalama Yakıt Sıcaklığı	654 °C	12	
HFP Etkin Yakıt Sıcaklığı	640 °C	12	
Kor Özellikleri:			
Toplam Yüklü Yakıt Kütlesi	81856 Ton	7	
Başlangıçtaki Yığın Sayısı	3	7	
Zenginliklerine göre her yığındaki yakıt demetleri			
	Birinci Döngü	İkinci Döngü	7,10,11
L	53 (2.1%)	9 (2.1%)	
M	52 (2.6%)	52 (2.6%)	
N	52 (3.1%)	52 (3.1%)	
P		44 (3.15%)	
Kor Çapı	152 cm		7
Yansıtıcı Özellikleri:			
Su Sıcaklığı	291.4 °C	7	
Su Basıncı	155 bar	7	
Kor Dış Kabı Materyali	SS-304	9	
Kor Dış Kabı Et Kalınlığı	2.857 cm	6	
Etkin Yansıtıcı Kalınlığı(radyal,eksenel)	25.2cm,38 cm	6	
Yakıt Demeti Özellikleri:			
Yakıt Demeti Sayısı	157	7	
Çubuk Dizilimi	17 x 17	6	
Her Demetteki Yakıt Çubuğu Sayısı	264	6	
Kontrol Çubuğu Kılavuz Kanalları:			
Materyal	Zircaloy-4	6	
Yoğunluk	6.55 g/cc	8	
Sayısı	24	6	
Dış Yarıçapı(Üst Bölgede, Amortisör Bölgesinde(51.0 cm))	0.612 cm, 0.545 cm	6	
Et Kalınlığı	0.0406 cm	6	
Zarf Malzemesi:			
Materyal	Zircaloy-4	6	
Yoğunluk	6.55 g/cc	8	
Dış Yarıçap	0.4750 cm	6	

İç Yarıçap	0.4179 cm	6
Zircolay Ağırlığı(zarf + kılavuz tüpü)	17320 x 10 ³ g	7
	Yakıt Tableti:	
Materyal	Sinterlenmiş UO ₂	6
Yoğunluk(yüzdece teorik)	95%	6
Yarıçap	0.4096 cm	6
Tablet Uzunluğu	1.346 cm	6
UO ₂ Yakıt Çubuğu	365.76 cm	6
Yüksekliği		
İç He Basıncı	450 psig	6
	Yanabilen Yutucu Çubuklar:	
Yutucuyu Bulunduran Materyal	Pyrex-glass	6
Materyaldeki oranca B miktarı(B ₂ O ₃)	12.5 w/o	6
Birim uzunlukta çubuktaki B-10 Kütlesi	0.006234 g/cm	6
Aktif Uzunluk	359.562 cm	6
Dış Yarıçap	0.48387 cm	6
Zarf Kalınlığı	0.04699 cm	6
Zarf Materyali	SS-304	6
İç Tüp Materyali	SS-304	6
İç Tüp Dış Yarıçapı	0.2305 cm	6
İç Tüp Kalınlığı	0.01651 cm	8
	Kontrol Çubukları:	
Materyaller:		
Yutucu	Ag (80%)- In (15%)-Cd (5%)	6
Zarf	SS-304	6
Yutucu Materyalinin:		
Yoğunluğu	10.1564 g/cc	6
Yarıçapı	0.43307 cm	6
Aktif Uzunluğu	360.7 cm	7
Zarf Materyalinin:		
Dış Yarıçapı	0.48387 cm	7
Kalınlığı	0.04699 cm	6
Kümedeki Kontrol Pin Sayısı	24	6

Almaraz NGS reaktör kor geometrisi aşağıda Şekil 2.2' de detaylı olarak gösterilmektedir. Geometrik yapının anlaşılması yazılan bilgisayar programının nasıl bir görsel sunuma sahip olması gerektiğinin taslağını çıkarmak açısından çok önemlidir. Çünkü hazırlanan grafiksel kullanıcı arayüzlerinde reaktör kor geometrisinin 1/8 'i esas alınmıştır.



Şekil 2.1 Reaktör kor geometrisi [5]

Şekil 2.2 'de kullanılan ifadelerin anlamları ve değerleri:

Inlet: Giriş

Outlet: Çıkış

A = 2.75 in (Pin 4 lokasyonu)

F= Reaktör kabı

B= 133.875 in (İç çap x 2.00 in Duvar)

G= 155.50 in (Kabın iç çapı)

C= Kor baffle

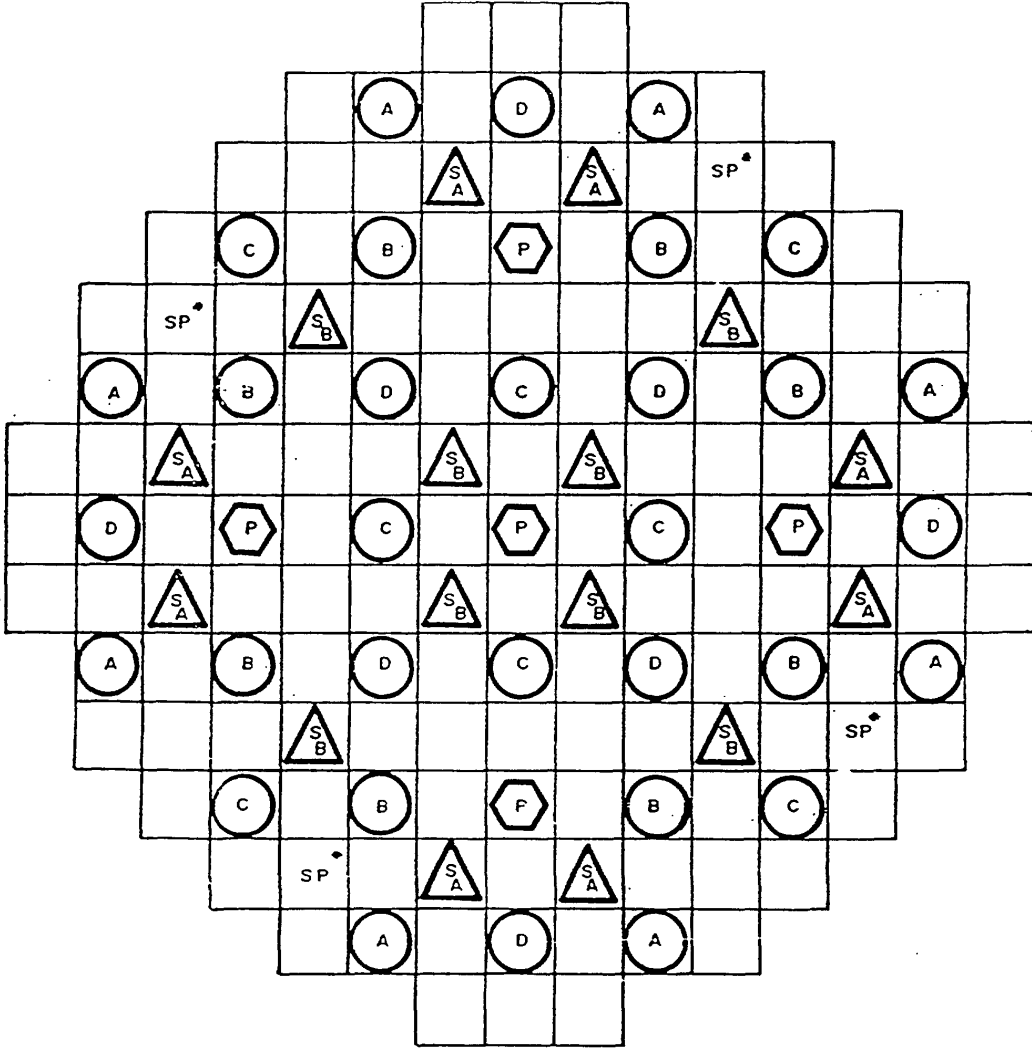
H= Kor varili(Kabı)

D=142.625 in (İç çap x 2.687 in Duvar)

I= Yaktı demeti

E= Termal zırh

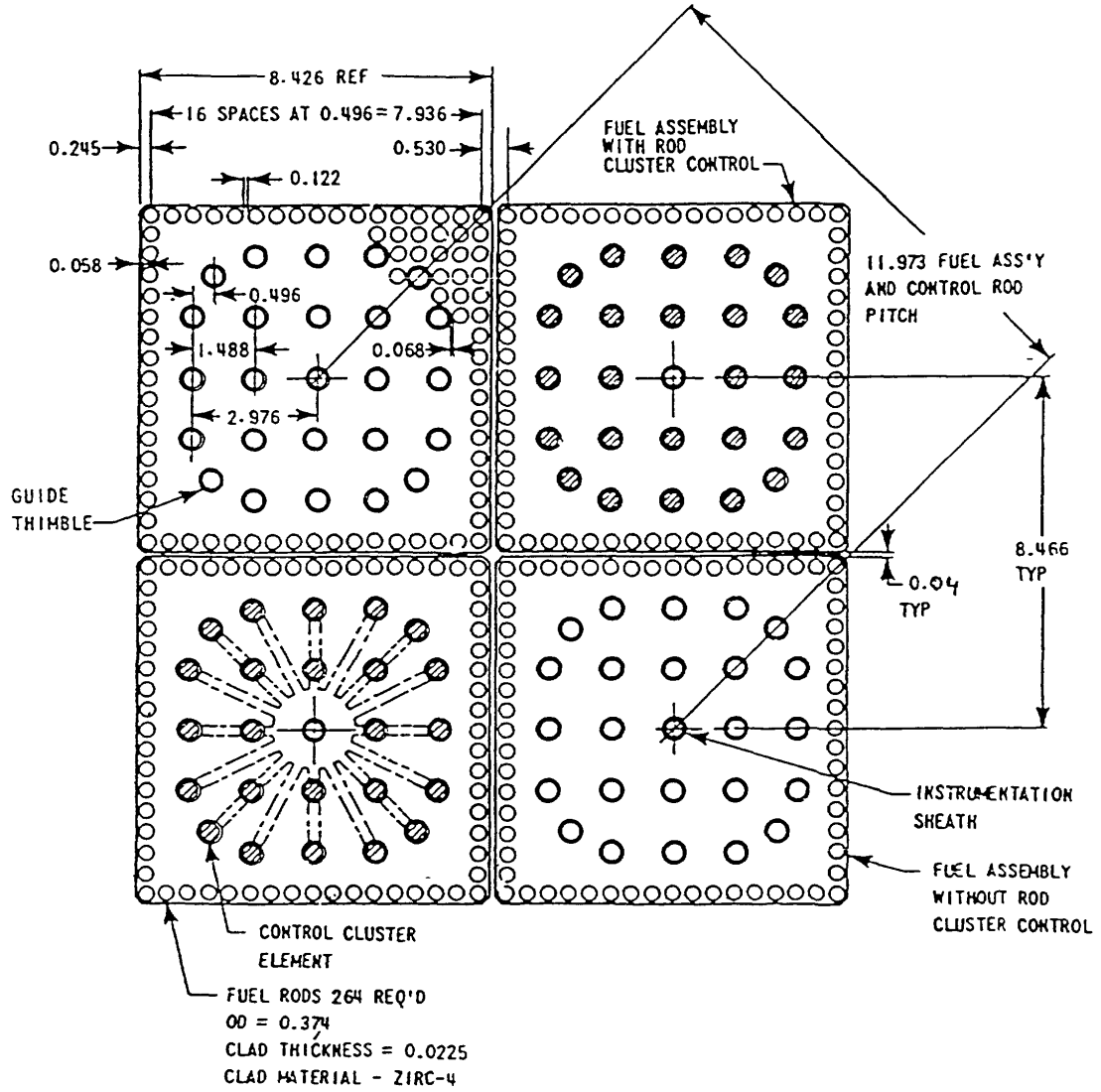
J= Su ile dolu alan



Şekil 2.2 Almaraz NGS çubuk kümesi ve kontrol çubukları demeti [5]

Kontrol bankları: A, B, C, D

Reaktör kapatma bankları: SA, SB



Şekil 2.3 Almaraz NGS yakıt demeti kesit görüntüsü 17 x 17 [5]

Şekil 2.4 te kullanılan kelimelerin anlamları:

Guide thimble: Klavuz tüpü (Kontrol çubuğu için)

Baffle: Bölme duvarı

Control cluster element: Kontrol demeti elemanı

Control rod pitch: Kontrol çubuğu açıklığı

Space: Boşluk

Instrumentation sheath: Enstrümantasyon kanalı

3. GELİŞTİRME ARAÇLARI

Hazırlanan grafiksek kullanıcı ara-yüzü için çeşitli programlama araçları ve kütüphaneleri kullanılmıştır. Kısaca bunlardan bahsetmek gerekirse FORTRAN programlama dili, Python programlama dili, aracı ve son olarak Qt kütüphanesi kullanılmıştır.

3.1. FORTRAN Programla Dili

En eski programlama dillerinden biri olan FORTRAN liderliğini Jhon Backus'un yaptığı IBM takımı tarafından ilk olarak 1957 yılında yayınlanmıştır. İsmi FORMula-TRANSlation (formül çevirme) kelimelerinin baş heceleri kullanılarak türetilmiştir. Bilimsel bir programlama dili olarak bilinen FORTRAN, ilk yüksek seviyeli derleyici ile sunulmuş programlama dilidir. Öncesinde program geliştiricilerinin kullandıkları donanıma bağlı olarak makine dili (assembly language) kullanmaları gerekiyordu. Makine dili yazım açısından oldukça zor, zaman alan, debug(hata ayıklaması) ve matematiksel ifadelerin dile çevrilmesi en büyük sıkıntılarıdır. FORTRAN programlama dili ile birlikte programları yazma hızı %500 artmış, kolay öğrenildiği için çok daha fazla program çok kısa sürelerde yazılmıştır. Bu gelişmeler programcı/araştırmacı/geliştirici/ 'nin makine dili ile zaman kaybetmek yerine asıl probleme daha iyi odaklanmasını sağlamıştır. FORTRAN programlama dili sadece ilk yüksek seviyeli dil olmasıyla değil derleyicilere getirdiği yaklaşımlarla da çok kişi tarafından benimsenmiş ve bireysel ihtiyaçlar doğrultusunda birçok kişi tarafından farklı şekilde geliştirilmeye başlanmıştır. Bu gelişmeler FORTRAN programlama dilinin taşınabilirliğini sınırlandırmıştır. Sorunları ortadan kaldırmak amacıyla American National Standards Association (Amerikan Ulusal Standartlar Kurumu) yapılan başvuru sonucu programlama dili bir standarda kavuşmuş ve 1966 yılında FORTRAN-66 (FORTRAN-IV) olarak yayınlanmıştır. Programla dili üzerindeki geliştirmeler devam ettiğinden 1978 de (FORTRAN-77), 1990' da (FORTRAN-90) şeklinde yeni gelişimleriyle birlikte standartları belirlenmiştir.

FORTRAN dilin öne çıkan özelliklerini kısaca şöyle sıralayabiliriz:

- 1) **Kolay öğrenilebilir:** dil kolayca öğrenilip, yazılabilir.
- 2) **Makineden bağımsız:** geliştirilen program makineden bağımsız olarak taşınabilir farklı niteliklerdeki makinelerde kolayca derlenip çalıştırılabilir.
- 3) **Probleme uyarlanabilir bir dil olması**

- 4) **Verimli çalışma** : makine diline kıyasla %20 oranında yavaş olmasına rağmen çok hızlıdır.
- 5) **Hafızayı kontrol etme yeteneği**: programı yazarken programcının Memory(RAM-Hafıza) üzerinde düşünmesine gerek yoktur.
- 6) **Program yazımında daha fazla özgürlük**: programı yazarken programcının satır sütunlar da kesin olarak uyması gereken kurallar çok azdır.

Sayılan özelliklerden dolayı FORTRAN programlama dili pek çok alanda etkin olarak kullanılmaktadır.[6]

Bu çalışmada geliştirilen tüm problemlerin esas çözümleri Fortran dilinde yazılmış olan programlarla çözülmüştür. Problemlerin matematiksel modellerinin uygulaması FORTRAN programlama dili kullanılarak sağlanmıştır. Bunun tercih edilmesinin sebebi geçmişten günümüze adaptasyonun kolay olması ve çok büyük işlemleri çok kısa sürede çözebilmesidir.

Geliştirilen fortran programları free-typing formatında yazılmış olup fortran-90 standartlarını taşımaktadır.

3.2. Python Programlama Dili

Python programlama dili 1980'lerin sonlarından itibaren CWI 'de (Centrum Wiskunde & Informatica- National Research Institute for Mathematics and Computer Science- Matematik ve Bilgisayar Bilimi Ulusal Araştırma Enstitüsü) Guido van Rossum tarafından Amoeba (operating system-işletim sistemi) için geliştirilmeye başlanan yüksek seviyeli bir programlama dilidir [7].

Python çok güçlü, dinamik bir programlama dili olmanın yanı sıra pek çok uygulama alanını da içerisinde barındırmaktadır. Kısaca etkin anahtar niteliklerinden bahsetmek gerekirse:

- 1) temiz ve okunabilir sözdizimine sahip olması,
- 2) güçlü iç kontrol mekanizmasına sahip olması,
- 3) sezgisel nesne yönelimine sahip olması,
- 4) usul kodlar için doğal sözdizimlerine sahip olması,

- 5) hiyerarşik yapıları destekleyen tamamen modüler bir yapıya sahip olması,
- 6) istisna tabanlı hata işleme yeteneğinin olması,
- 7) çok yüksek seviyeli dinamik veri tiplerine sahip olması,
- 8) hemen hemen her görev için gerekli modüllere ve standart kütüphaneye sahip olması,
- 9) uzantıları ve modülleri C, C++, Java, FORTRAN gibi her dilde yazılabilir olması,
- 10) bir komut dosyası arabirimi gibi uygulamalar içine gömülebilmesi sayılabilir [8].

Bu özelliklerinden dolayı Python programlama dili gerekli grafiksel ara-yüzün oluşturulması ve bunun FORTRAN - Qt kütüphanesiyle köprü olarak kullanılması açısından fazlasıyla kullanışlıdır.

3.3. Qt Kütüphanesi

Qt çapraz platformlarda(MS Windows, Linux/GNU, MacOSX, iOS, Android gibi) çalışabilen C++ programı ile geliştirilmiş olan bir kullanıcı arayüzü geliştirme kütüphanesidir.[9] Qt kütüphanesi için tüm tanımlamalar C++ dilinde yapılmış ve diğer programlama dilleri içinde uyarlamaları geliştirilmiştir. Bu çalışmada Qt kütüphanesinin python diline uyarlanmış kütüphanesi PyQt ve PySide kullanılmıştır. Qt kütüphanesinin kullanılma amacı:

- 1) doğal pencereler kullanması,
- 2) çalıştığı platformun doğal temalarını desteklemesi,
- 3) arayüz geliştirme araçlarına sahip olması,
- 4) kolay-anlaşılır sözdizimine sahip olması,
- 5) çapraz-platformlarda çalışabilmesi,

şeklinde özetlenebilir.

4. MATEMATİKSEL MODELLER

Nükleer reaktörlerde oluşan temel reaksiyon fisyon tepkimesidir. Fisyon tepkimesi sonucu açığa çıkan nötronlar tekrar yutulurken yenilerinin doğmasına sebep olurlar, bu olay zincirleme olarak devam eder.

Nükleer reaktörlerde belli bir konumdaki ve belli enerjideki nötronların zamana göre sayısının tayini için nötron transport denklemi kullanılabilir. Bu denklem aşağıdaki gibidir [10].

$$\begin{aligned} & \frac{1}{v} \frac{\partial}{\partial t} \psi(r, E, \underline{\Omega}, t) + \underline{\Omega} \cdot \nabla \psi(r, E, \underline{\Omega}, t) + \Sigma_t(r, E) \psi(r, E, \underline{\Omega}, t) \\ & = \int_{4\pi} d\underline{\Omega} \int_0^\infty dE' \Sigma_s(r, E \rightarrow E', \underline{\Omega} \rightarrow \underline{\Omega}') \psi(r, E, \underline{\Omega}, t) \\ & + \frac{\chi_p(E)(1-\beta)}{4\pi} \int_{4\pi} d\underline{\Omega} \int_0^\infty dE' \nu \Sigma_f(r, E', \underline{\Omega}) \psi(r, E', \underline{\Omega}, t) \\ & + \frac{1}{4\pi} \beta \sum_i \chi_i(E) \lambda_i C_i(r, t) + s(r, E, \underline{\Omega}, t) \end{aligned} \quad 4.1$$
$$\frac{\partial}{\partial t} C_i(r, t) = -\lambda C_i(r, t) + \int_{4\pi} d\underline{\Omega} \int_0^\infty dE' \beta_i(E') \nu \Sigma_f(r, E', \underline{\Omega}, t)$$

Denklem içerisindeki ifadelerin anlamlarını yazmak gerekirse:

v	Nötronun hızı(cm/s),
t	Zaman(s),
$r \in \mathbb{R}^3$	Konum vektörü,
$\underline{\Omega} \in \mathbb{R}^2$	Açısal yönelim vektörü,
E	Enerji(MeV),
Σ_t	Toplam tesir kesiti,
Σ_s	Saçılma tesir kesiti,
Σ_f	Fisyon tesir kesiti,
ν	Fisyon başına çıkan ortalama nötron sayısı,
χ_p	Anlık (prompt) nötronların enerji spektrumu,
χ_i	i. inci gecikmiş (delayed) nötron grubunun enerji spektrumu,

C_i	i. inci öncülerin konsantrasyonu,
λ_i	i. inci çekirdeğin bozunum katsayısı,
β_i	i. inci gecikmiş nötronların kesri,
ψ	nötron akısı.

Nötron transport denkleminin karmaşık ve büyük sistemler için çözümü zor ve maliyetli olduğundan difüzyon yöntemi kullanılmıştır. Yazılan bilgisayar programları için adapte edilen yaklaşım bir-buçuk grup nötron difüzyon denklemdir.

4.1. Bir-buçuk Grup Nötron Difüzyon Denklemi Yaklaşımı

Bir-buçuk grup nötron difüzyon denklemi, taşınım denkleminin iki grup nötron (fast-thermal/ hızlı-termal) için yakınsanmış halidir. Denklemi aşağıdaki gibi yazıp düzenlemek gerekirse;

Hızlı grup için:

$$-D_1 \nabla^2 \phi_1 + \Sigma_{a1} \phi_1 + \Sigma_{12} \phi_1 - \frac{1}{k} (\nu \Sigma_{f1} \phi_1 + \nu \Sigma_{f2} \phi_2) = 0 \quad 4.1.1$$

Termal grup için:

$$-D_2 \nabla^2 \phi_2 + \Sigma_{a2} \phi_2 - \Sigma_{12} \phi_1 = 0 \quad 4.1.2$$

Burada D_1 ve D_2 hızlı ve termal grupların difüzyon katsayılarıdır. Bir-buçuk grup nötron difüzyon denkleminde termal nötronlardan kaynaklanan sızıntı ihmal edilmektedir. Bu oran %5 ten küçüktür.[11]

Bu varsayımla denklem yeniden düzenlenecek olursa:

$$\phi_2 = \frac{\Sigma_{12}}{\Sigma_{a2}} \phi_1 \quad 4.1.3$$

$$-D_1 \nabla^2 \phi_1 + \Sigma_{a1} \phi_1 + \Sigma_{12} \phi_1 - \frac{1}{k} (\nu \Sigma_{f1} \phi_1 + \nu \Sigma_{f2} \frac{\Sigma_{12}}{\Sigma_{a2}} \phi_1) = 0 \quad 4.1.4$$

Nötron göç alanını (migration area) tanımlayacak olursak:

$$M^2 = \frac{D_1}{\Sigma_{a1} + \Sigma_{12}} \quad 4.1.5$$

sonsuz ortam çarpan faktörü ise,

$$k_\infty = \frac{\nu \Sigma_{f1} + \nu \Sigma_{f2} \frac{\Sigma_{12}}{\Sigma_{a2}}}{\Sigma_{a1} + \Sigma_{12}} \quad 4.1.6$$

şeklinde ifade edilebilir. Denklem 4.1.4 ü tekrar düzenlersek,

$$-D_1 \nabla^2 \phi_1 + \left(\frac{k_\infty}{k} - 1 \right) \frac{1}{M^2} \phi_1 = 0 \quad 4.1.7$$

buna göre bölgesel reaktiviteyi:

$$\rho_\infty = 1 - \frac{1}{k_\infty} \quad 4.1.8$$

ρ_∞ değeri ρ olarak tanımlanırsa sistem reaktivitesi:

$$\rho_s = 1 - \frac{1}{k} \quad 4.1.9$$

şeklinde tanımlanır.

Yeni değişkenlerle birlikte denklem 4.1.7 yi yeniden yazarsak:

$$\nabla^2 \phi_1 + \left(\frac{\rho - \rho_s}{1 - \rho} \right) \frac{1}{M^2} \phi_1 = 0 \quad 4.1.10$$

Elde edilen denklem için reaktörün kritik olduğu varsayılırsa ($k = 1$) olacağından sistem reaktivitesi sıfır olacaktır. Buna göre;

$$\nabla^2 \phi_1 + \left(\frac{\rho}{1-\rho} \right) \frac{1}{M^2} \phi_1 = 0 \quad 4.1.11$$

denklemini elde edilir.

Nükleer reaktörlerde ki fisyon reaksiyonu sonucu ortaya çıkan enerjinin nötron akısıyla ilişkisini bir-buçuk grup nötron difüzyon denklemine göre yazmak gerekirse;

$$q''' = \kappa \Sigma_{f1} \phi_1 + \kappa \Sigma_{f2} \phi_2 \quad 4.1.12$$

buradaki κ sembolü fisyon başına ortaya çıkan enerji miktarıdır.

$$q''' = \kappa \left(\Sigma_{f1} \phi_1 + \Sigma_{f2} \frac{\Sigma_{12}}{\Sigma_{a2}} \phi_1 \right) \quad 4.1.13$$

tekrar açık haliyle düzenlenirse:

Denklem 4.1.14 düzenle v koy (-) leri + yap

$$\begin{aligned} q''' &= \kappa \left(\Sigma_{f1} \phi_1 + \Sigma_{f2} \frac{\Sigma_{12}}{\Sigma_{a2}} \phi_1 \right) \frac{\nu \Sigma_{a1} + \Sigma_{12}}{\nu \Sigma_{a1} + \Sigma_{12}} \frac{D_1}{D_1} \\ &= \kappa \frac{1}{\nu} \phi_1 \frac{\left(\Sigma_{f1} + \Sigma_{f2} \frac{\Sigma_{12}}{\Sigma_{a2}} \right) \Sigma_{a1} + \Sigma_{12}}{\Sigma_{a1} - \Sigma_{12}} \frac{D_1}{D_1} \nu \end{aligned} \quad 4.1.14$$

$$\begin{aligned} q''' &= \kappa \frac{1}{\nu} \phi_1 k_{\infty} \frac{1}{M^2} D_1 \\ &= \frac{\kappa}{\nu} k_{\infty} \frac{D_1}{M^2} \phi_1 \end{aligned} \quad 4.1.15$$

Denklem 4.1.15 ten hızlı nötron akısı çekilirse,

$$\phi_1 = q''' \frac{V}{\kappa} \frac{M^2}{D_1} \frac{1}{k_\infty} \quad 4.1.16$$

ya da eşdeğer olarak,

$$\phi_1 = q''' \frac{V}{\kappa} \frac{M^2}{D_1} (1 - \rho) \quad 4.1.17$$

şeklinde olduğu görülür.

N toplam düğüm noktası, V hacim, Q reaktörün termal gücü, f_i i noktasındaki tüm güç üzerinden ortalama güç çarpanı değeri olarak alınırsa,

$$f_i = \frac{1}{Q} \int_V q''' dV = q''' \frac{N}{Q} V \quad 4.1.18$$

denklemini elde edilir.

Denklem 4.1.18 ile denklem 4.1.11 ilişkilendirilerek tekrar düzenlenirse,

$$\nabla^2 [q'''(1 - \rho)] + \frac{\rho q'''}{M^2} = 0 \quad 4.1.19$$

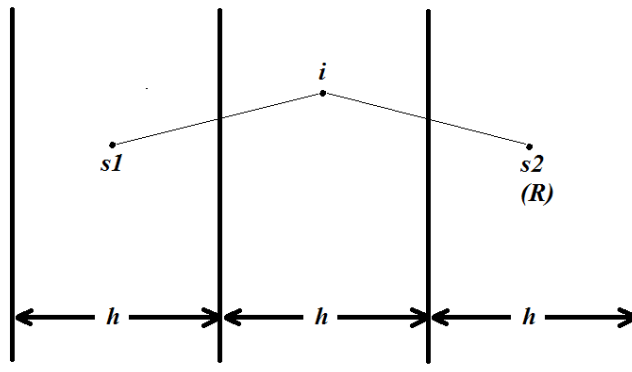
elde edilir bu denklemini de reaktörün toplam termal gücü üzerinden normalize edilirse,

$$\nabla^2 [f_i(1 - \rho_i)] + \frac{\rho_i f_i}{M^2} = 0 \quad 4.1.20$$

denklemini elde edilmiş olur. Bu denklem bilgisayar programı yazılırken analitik eşitliğin nümerik bir eşitliğe dönüşmüş hali olarak programda kullanılır.[12]

4.2. Levha Geometrisi için Güç Paylaşım Denkliklerinin Türetilmesi [11]

İki komşu düğüm noktası tarafından şekil 4.1'deki gibi sarılmış olan bir i düğüm noktası için yaklaşım aşağıdaki gibidir. Etrafında $s1$ ve $s2$ yakıt demetleri olduğu durumda, bir iç düğüm noktasını, $s2$ değerinin yansıtıcı bir yüzeye eşit olduğu durumda ise R ise çevresel bir düğüm noktasını ifade etmektedir.



Şekil 4.1 Levha Geometrisi

Sonlu farklar yöntemine göre iç düğüm noktaları kullanılarak denklem 4.1.11 yazılırsa ve ρ_{∞} değeri ρ olarak tanımlanırsa;

$$\frac{\phi_{s1} - 2\phi_i + \phi_{s2}}{h^2} + \left(\frac{\rho}{1-\rho}\right) \frac{1}{M^2} \phi_i = 0 \quad 4.2.1$$

Nötron akısı yerine yerel güç değeri konulursa,

$$\frac{q_{s1}(1-\rho_{s1}) - 2q_i(1-\rho_i) + q_{s2}(1-\rho_{s2})}{h^2} + \left(\frac{\rho}{1-\rho}\right) \frac{1}{M^2} q_i(1-\rho_i) = 0 \quad 4.2.2$$

denklemini yeniden düzenleyecek olursak,

$$\frac{q_{s1} + q_{s2}}{2} - \frac{q_{s1}\rho_{s1} + q_{s2}\rho_{s2}}{q_{s1} + q_{s2}} \frac{q_{s1} + q_{s2}}{2} + q_i \left(1 - \rho_i - \frac{1}{2} \frac{h^2}{M^2} \rho_i \right) = 0 \quad 4.2.3$$

burada $\frac{q_{s1} + q_{s2}}{2}$ 'yi ortalama \bar{q} , $\frac{q_{s1}\rho_{s1} + q_{s2}\rho_{s2}}{q_{s1} + q_{s2}}$ değeri için ise güç değerleri üzerinden normal alınmış $\bar{\rho}$ olarak tanımlanırsa, etrafında yakıt demetleri olan bir i . nci düğüm noktası için güç yoğunluğu:

$$q_i = \frac{\bar{q}(1 - \rho_i)}{1 - \theta\rho_i} \quad 4.2.4$$

şeklinde bulunur. Burada $\theta = 1 + \frac{1h^2}{2M^2}$ ' dir.

Sistemin yüzeyinde yansıtıcı olan kısmı için çözüm yapmak gerekirse ϕ_{s2} yerel güç akısı yerine ϕ_R alınır ve sınırda albedo sınır koşulu kullanılırsa çözüm aşağıdaki şekli alır.

$$\alpha = \frac{J^-}{J^+} = \frac{\phi + 2D \frac{d\phi}{dx}}{\phi - 2D \frac{d\phi}{dx}} \quad 4.2.5$$

tam yüzey noktasındaki yaklaşık değeri alınır,

$$\frac{1}{\phi} \frac{d\phi}{dx} \sim \frac{1}{(\phi_R - \phi_i)} \frac{(\phi_R - \phi_i)}{h} \quad 4.2.6$$

Bu yaklaşım ile denklem 4.2.5 yeniden yazılacak olursa,

$$\alpha = \frac{1 + 2D \frac{1}{\phi} \frac{d\phi}{dx}}{1 - 2D \frac{1}{\phi} \frac{d\phi}{dx}} = \frac{1 + \frac{4D}{h} \frac{\phi_R - \phi_i}{\phi_R + \phi_i}}{1 - \frac{4D}{h} \frac{\phi_R - \phi_i}{\phi_R + \phi_i}} \quad 4.2.7$$

$$= \frac{(\phi_R + \phi_i) + \frac{4D}{h}(\phi_R - \phi_i)}{(\phi_R + \phi_i) - \frac{4D}{h}(\phi_R - \phi_i)}$$

ϕ_R için denklem düzenlenecek olursa,

$$\phi_R = \left[\frac{1 - \frac{(1-\alpha)h}{(1+\alpha)4D}}{1 + \frac{(1-\alpha)h}{(1+\alpha)4D}} \right] \phi_i = \gamma \phi_i \quad 4.2.8$$

Yansıtıcı bölge için oluşturulan bu akı değeri denklem 4.2.1 ' de yerine konulursa,

$$\phi_{s1} - 2\phi_i + \gamma\phi_i + \left(\frac{\rho_i}{1-\rho_i} \right) \frac{h^2}{M^2} \phi_i = 0 \quad 4.2.9$$

Denklem 4.2.8 ile denklem 4.2.4 yeniden düzenlenirse,

$$q_i = \frac{\bar{q}(1-\rho_i)}{1-\theta_p(\rho_i-\rho_L)} \quad 4.2.10$$

burada,

$$\theta_p = 2\theta - \gamma = \frac{2\theta - 1}{1 - \rho_L} \quad 4.2.11$$

sızıntı reaktivitesini denklemden çekersek,

$$\rho_L = \frac{1-\gamma}{2\theta-\gamma} \quad 4.2.12$$

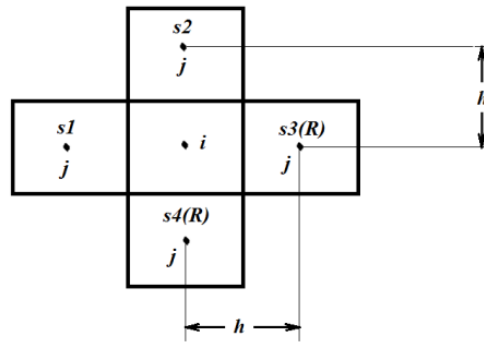
şeklinde levha geometrisi için güç yoğunluğu denklemlerini bilgisayar programına uyarlayabilecek şekilde elde etmiş oluruz.[12]

4.3. Kare Şeklindeki Düğüm Noktaları için Güç Paylaşım Denkliklerinin Türetilmesi [11]

Kritik bir sistem için akı bükülmesi yönelimler boyunca eşit alınacak olursa, sistemin çözümü için aşağıdaki ifade kullanılır.

$$B_M^2 = \left(\frac{\rho}{1-\rho} \right) \frac{1}{M^2} = B_g^2 = B_x^2 + B_y^2 \quad 4.3.1$$

Kartezyen geometri için sonlu farklar yöntemi sisteme uygulanacak olursa,



Şekil 4.2 Kartezyen geometri için düğüm noktaları [12]

S, i düğüm noktası için komşuluklarındaki yakıt demetlerinin sayısı olmak üzere, R i düğüm noktasını çevreleyen yansıtıcı yakıt demetlerinin sayısı olmak üzere sonlu farklar yöntemi kare-kartezyen geometri için 4.2 başlığındaki denklemler yeniden uygulanırsa aşağıdaki denklemler sırası ile elde edilir.

$$\theta_0 = 1 + \frac{1}{4} \frac{h^2}{M^2} \quad 4.3.2$$

$$\sum_{j=1}^S \phi_{sj} + 4\phi_i + (4-S)\phi_i \gamma = -\frac{h^2}{M^2} \left(\frac{\rho_i}{1-\rho_i} \right) \phi_i \quad 4.3.3$$

$$q_i = \frac{\frac{1}{S} \sum_{j=1}^S q_{sj} (1-\rho_{sj})}{\frac{1}{S} \left([4-(4-S)\gamma](1-\rho_i) - \frac{h^2}{M^2} \rho_i \right)} \quad 4.3.4$$

$$\begin{aligned}
4 - S &= R \\
\frac{1}{S} \left((4 - R\gamma)(1 - \rho_i) - \frac{h^2}{M^2} \rho_i \right) \\
\frac{1}{S} \left(4 - R\gamma + R\gamma\rho_i - 4\rho_i - \frac{h^2}{M^2} \rho_i \right) \\
\frac{1}{S} (4 - R\gamma + R\gamma\rho_i - 4\theta_0\rho_i) \\
\frac{1}{S} ((R + S) - R\gamma) - \frac{(4\theta_0 + R\gamma)}{S} \rho_i \\
1 + \frac{R(\gamma - 1)}{S} - \frac{(4\theta_0 + R\gamma)}{S} \rho_i
\end{aligned} \tag{4.3.5}$$

$$\theta_R = \frac{((R + S)\theta_0 + R\gamma)}{S} \tag{4.3.6}$$

$$\rho_L = \frac{\frac{R(\gamma - 1)}{S}}{\theta_R} \tag{4.3.7}$$

$$q_i = \frac{\frac{1}{S} \sum_{j=1}^S q_{sj} (1 - \rho_{sj})}{1 - \theta_R (\rho_i - \rho_L)} \tag{4.3.8}$$

Yapılan yaklaşımlar ile sistemin tamamı için iki boyutlu olarak güç yoğunluğu denklileri çıkartılmıştır. Bu denklilerin FORTRAN programlama diline adaptasyonu sağlanarak sistem için program yazılmıştır. [12]

5. PROGRAMLAMADA GENETİK ALGORİTMA YAKLAŞIMI

Genetik Algoritmalar, geleneksel optimizasyon metotlarına alternatif olarak görülen bir evrimsel optimizasyon yaklaşımıdır. GA, global optimum noktasının yerinin bulunmasının zor olduğu karmaşık doğrusal olmayan modellerin çözümü için çok uygundur. Bu algoritmalar biyolojik organizmaların genetik sürecine dayanır.

Genetik Algoritma yaklaşımının ortaya çıkışı 1970'lerin başında olmuştur. 1975'te John Holland'ın makine öğrenmesi üzerine yaptığı çalışmalarda canlılardaki evrimden ve değişimden etkilenerek, bu genetik evrim sürecini bilgisayar ortamına aktarması ve böylece bir tek mekanik yapının öğrenme yeteneğini geliştirmek yerine, çok sayıdaki böyle yapıların tamamını "çiftleşme, çoğalma, değişim..." gibi genetik süreçler sonunda üstün yeni bireylerin elde edilebileceğini gösteren çalışmasından çıkan sonuçların yayınlanmasından sonra geliştirdiği yöntemin adı "Genetik Algoritmalar" olarak tanınmıştır [13].

Genetik Algoritmalar doğadaki canlıların geçirdiği evrim sürecini örnek alarak matematiksel modeli kurulamayan ya da çözüm uzayı çok geniş olan problemlerin çözümünde tercih edilen tekniklerin başında gelmektedir.

GA'nın evrimden yararlanma düşüncesi, bir bireyin hem annesinin hem de babasının özelliklerini taşıyabildiği gibi onlardan daha farklı ve üstün özelliklerini de taşıyabileceği varsayımına dayanmaktadır.

GA da her bir çözüm, birey veya kromozom adı verilen dizinlerle gösterilir. Biyolojiden esinlenerek genellikle 0 ve 1'lerden oluşan dizinlerle ifade edilirler. Biyolojik kromozom üzerinde belirli genlerin belirli karakteristik özellikleri taşıması gibi genetik algoritmaların belirli kısımlarının da belirli özellikleriyle problemin çözümünü içerdiği kabul edilir. Son yıllarda üretim planlama, tasarım, elektronik, finansman gibi farklı, çok geniş alanları kapsayan konularda GA çalışmaları hız kazanmış, bu ilgi giderek her geçen gün artmaktadır [14].

Bu yaklaşım doğrultusunda nükleer yakıt yönetim optimizasyonu GA'nın birer permütasyon problemi olarak ortaya çıkar. GA fonksiyonlarından çaprazlama fonksiyonu kullanılarak kromozomların oluşturulması sağlanır [15]. (Burada oluşturulan kromozomlar hali hazırda çözüm kümesi olarak sahip olunan nükleer yakıt demetleri tipleridir.) Başlangıç olarak bu kromozomlar bir boyutlu bir dizi olarak yakıt demetlerinin tiplerine göre oluşturulur [16].

Oluşturulan kromozomlar her yeni bir yakıt düzeni için yenilenerek çoğaltılır. Bu çözümler doğrultusunda nükleer reaktörün yakıt yüklemesi için en uygun yükleme düzeni belirlenen koşullar altında program tarafından oluşturulmaya çalışılır.

6. Problemlerin Tanımlanması ve Grafiksel Kullanıcı Arayüzleri

Bu çalışmada daha önce de söz edildiği gibi Almaraz NGS için kor yükleme düzenlerini hesaplayan veya girilen kor yükleme düzenlerine göre güç dağılımlarını hesaplayan RPM (reloading power mapping/yeniden yükleme güç haritalaması) programları için grafiksel kullanıcı arayüzleri geliştirilmiştir. Bu matematiksel modellerin uygulanması kolaydan karmaşığa şeklinde düzenlenmiş ve ilk problem tek boyutlu bir kor geometrisinin çözümü için tasarlanmıştır. Buradaki amaç kullanıcının Almaraz NGS'nin kor yükleme düzenine geçmeden önce hazırlık yapmasını problemi ve kullanıcı arayüzünü tanımasını sağlamaktır. İlk problemde kullanıcının belirli üç tipteki basınçlı su reaktörü yakıtının yanma modelini tasarlaması ve bunların sonuçlarını gözlemesi hedeflenmiştir.

Tüm programlar Almaraz NGS Kor yükleme düzeni için geliştirilmiştir. Buna rağmen program yeni uygulamalara ve geliştirmelere açık olarak tasarlanmıştır. Grafiksel kullanıcı arayüzlerini tanıtmadan önce kısaca geliştirilmiş olunan RPM programından bahsetmek gerekir. Almaraz NGS için verilen kor yükleme düzeni 1/8 i olmak üzere aşağıdaki gibidir ve tüm çalışma boyunca bu yapının simetriği esas alınmıştır.

121 1					
120 2	121 3				
121 4	116 5	121 6			
120 7	121 8	116 9	121 10		
121 11	116 12	121 13	116 14	121 15	
112 16	121 17	116 18	312 19	131 20	
121 21	316 22	131 23	131 24		
131 25	131 26				

N
 P
N: fuel assembly ID
P: position index

Şekil 6.1 Almaraz Kor Yükleme Düzeni [17]

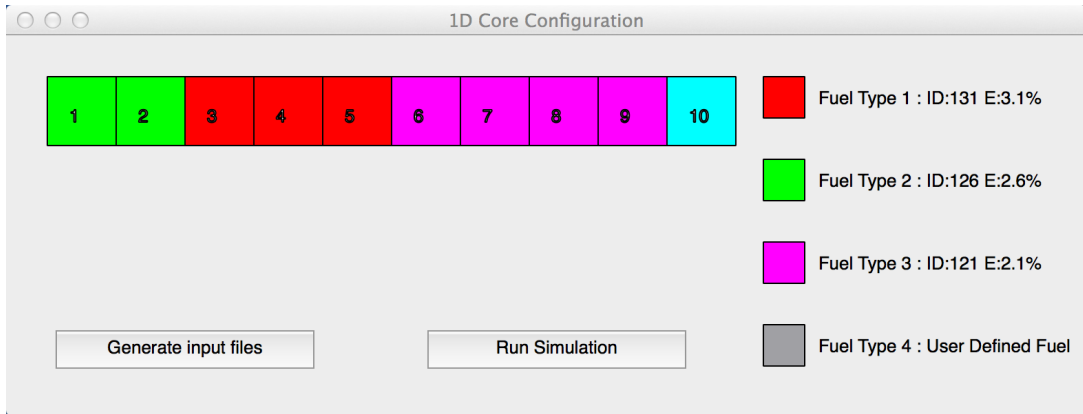
6.1. Tek Boyutlu ve İki Boyutlu Kor Yükleme Hesaplaması için RPM Programı ve Grafikselle AraYüz(PyRPM)

İlk problem olarak tasarlanan tek boyutlu bir reaktör korunun yakıt yükleme düzeni önceden belirlenmiş yakıtlar için bunların kor içindeki dizilimi ve yanma oranlarının hesaplamaları yapılmıştır. Bunun için temel bir grafikselle arayüz geliştirilmiştir. Ardından sızıntı reaktiviteleri bilinen bir yakıt yükleme düzeni için 4. başlıktaki denklemler kullanılarak Almaraz NGS için FORTRAN programı yazılmış ve bu programın bir grafikselle kullanıcı arayüzü oluşturulmuştur. Programın FORTRAN dili ile yazılmış olan kısmı girilen yakıt yükleme düzeni için sonlu farklar yöntemini kullanarak yanma oranlarını, boron konsantrasyonunu ve yakıt demetleri için güç değerlerini hesaplamaktadır. Program üç ana bölümden oluşmaktadır.

İlk bölüm yakıt yükleme düzenine ait bilgilerin grafikselle arayüz aracılığıyla oluşturulması, ikinci bölüm problemin FORTRAN programlama diliyle yazılmış programlarla çözülmesi, üçüncü ve son bölüm ise çıktı dosyalarının oluşturulmasıdır.

Programın sürecini ve giriş yöntemlerini şekiller ve program görüntüleriyle inceleyelim.

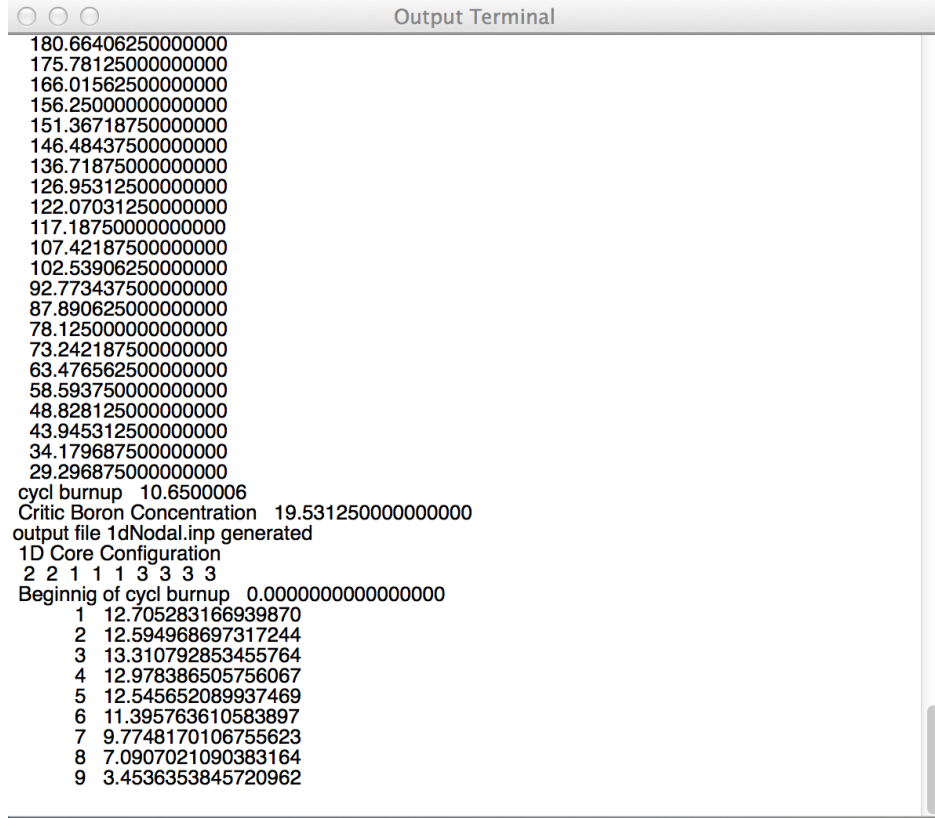
Tek boyutlu yakıt yükleme düzenini içeren programın grafikselle arayüzüne göz atarak başlamak gerekirse;



Şekil 6.2 1dNodal Programı arayüzü

Şekil 6.2 'de ki arayüz üzerinden dokuz adet yakıt hücresi için üç tip yakıttan biri ilgili kutucuğun üzerine tıklanarak seçilir. Seçimin tamamlanmasının ardından çıktı

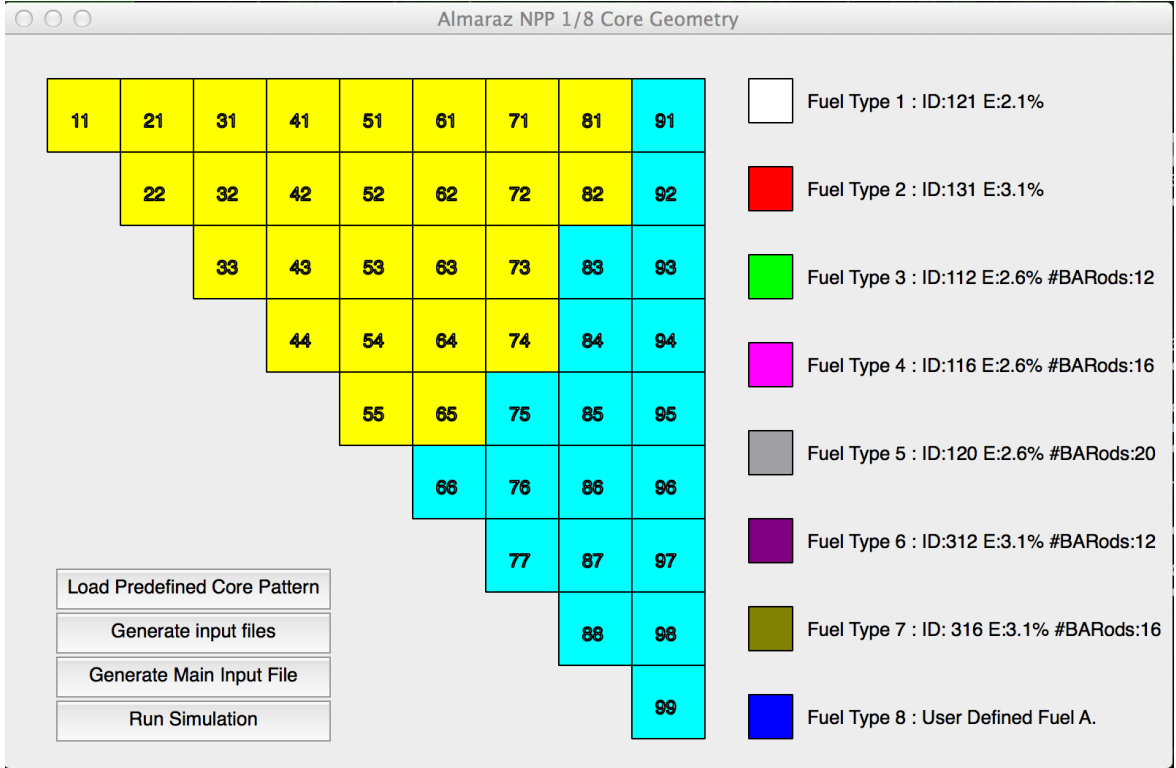
kontrol penceresi üzerinden işlemler takip edilerek sonuç kontrol edilir. Girdilerde herhangi bir sorun yoksa önce Generate Input Files butonuna basılarak gerekli girdi dosyası yaratılır ve ardından Run Simulation butonuna basılarak programın çalışması ve sonuçlarının ekrana yazdırılması sağlanır. Ekran çıktısı aşağıdaki şekilde olacaktır.



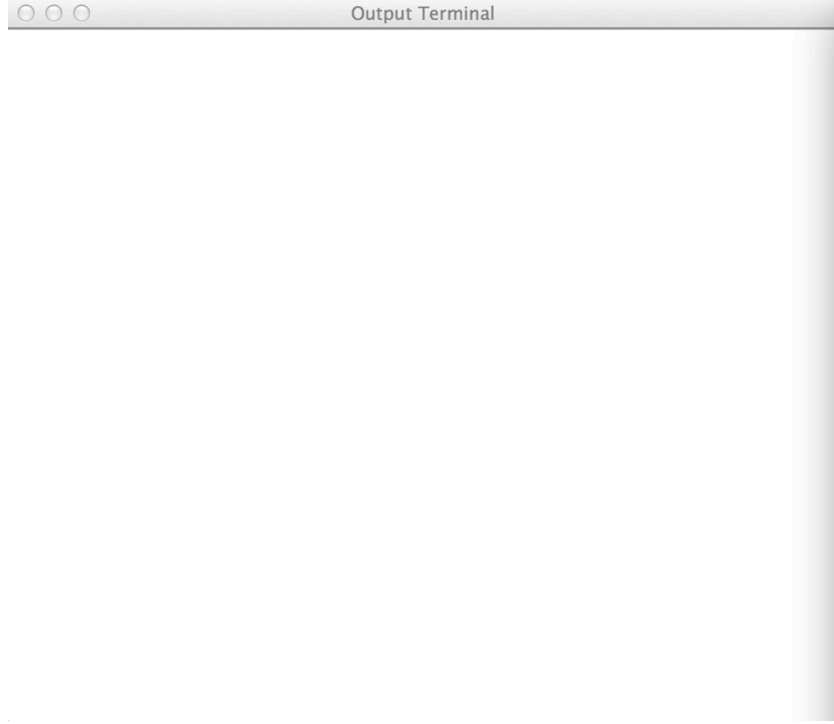
```
Output Terminal
180.66406250000000
175.78125000000000
166.01562500000000
156.25000000000000
151.36718750000000
146.48437500000000
136.71875000000000
126.95312500000000
122.07031250000000
117.18750000000000
107.42187500000000
102.53906250000000
92.77343750000000
87.89062500000000
78.12500000000000
73.24218750000000
63.47656250000000
58.59375000000000
48.82812500000000
43.94531250000000
34.17968750000000
29.29687500000000
cycl burnup 10.6500006
Critic Boron Concentration 19.531250000000000
output file 1dNodal.inp generated
1D Core Configuration
2 2 1 1 1 3 3 3 3
Beginnig of cycl burnup 0.0000000000000000
1 12.705283166939870
2 12.594968697317244
3 13.310792853455764
4 12.978386505756067
5 12.545652089937469
6 11.395763610583897
7 9.7748170106755623
8 7.0907021090383164
9 3.4536353845720962
```

Şekil 6.3 Çıktı terminali

Almaraz NGS modellemesinin grafik arayüzüne bakmak gerekirse burada tüm kor geometrisinin simetrik en küçük parçası alınarak tasarlanmıştır. Bu güç santrali için belirlenmiş yedi(7) tip farklı özelliklerde yakıt demeti mevcuttur.

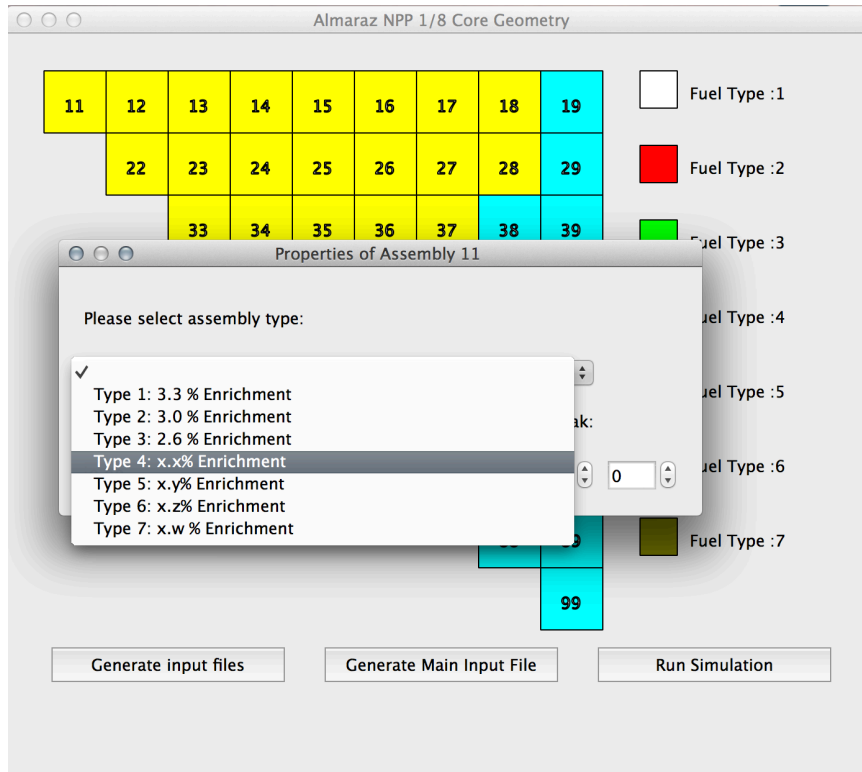


Şekil 6.4 PyRPM program ana düzenleme sayfası



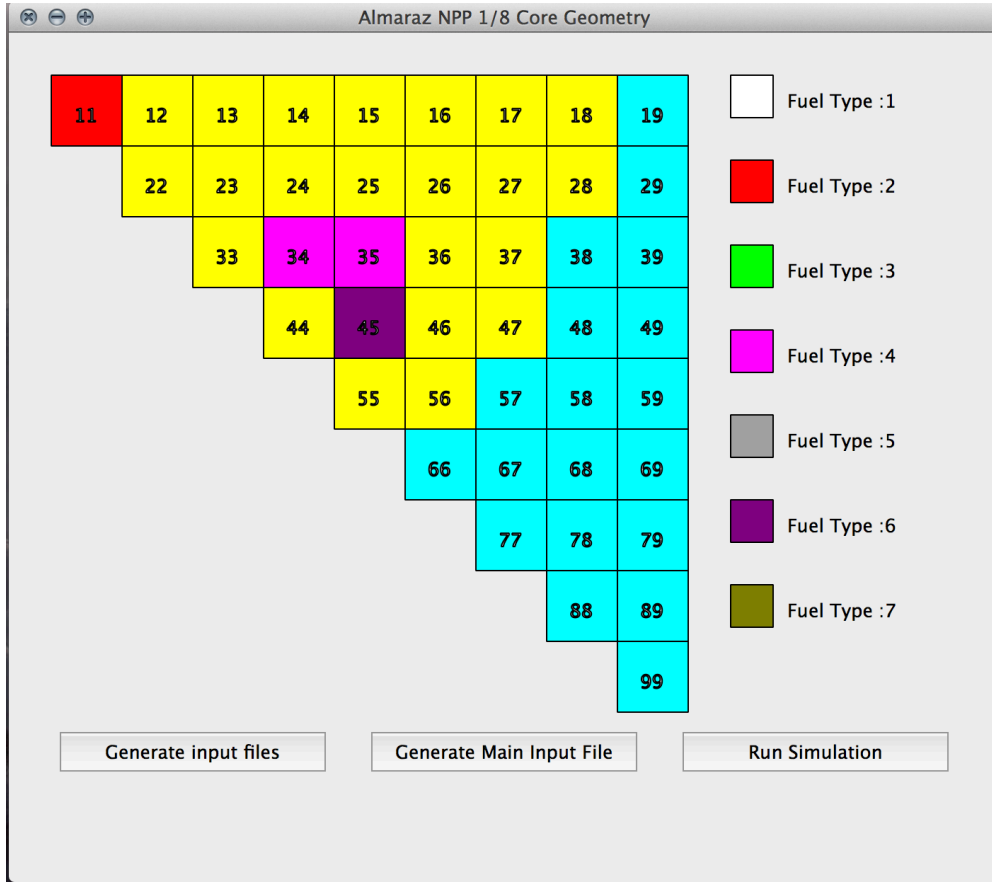
Şekil 6.5 PyRPM program ana çıktı sayfası

Programın açılmasıyla birlikte Python ve Qt kullanılarak hazırlanmış olan ana sayfalar ekrana gelmektedir. Ana sayfalardan şekil 6.4'te verilerin girileceği, şekil 6.5' de ise girilen verilen programdaki karşılığının çıktıları ekrana gelmektedir.



Şekil 6.6 Yakıt demetlerinin seçimi

Şekil 6.6'da görüldüğü gibi başlangıç kor yüklem düzeninin belirlenebilmesi için her yakıt çubuğuna fare imleci ile tıklanarak o yakıt demetinin tipinin belirtilmesi gerekir. Yakıt tipinin belirtilmesi ile birlikte Şekil 6.4'te görülen sayfanın sağ tarafındaki yakıtlardan biri seçilmiş olunur yanı sıra aynı ekrandan o yakıt çubuğuna ait sızıntı reaktivitesi ve kaç yüzeyden sızıntı yaptığı girilir. Sızıntı reaktivite değeri sadece açık-mavi(cyan) komşulukları için geçerlidir. Bunun sebebi alınan kesitin korun 1/8 olması ve simetri esasını korumaktır.



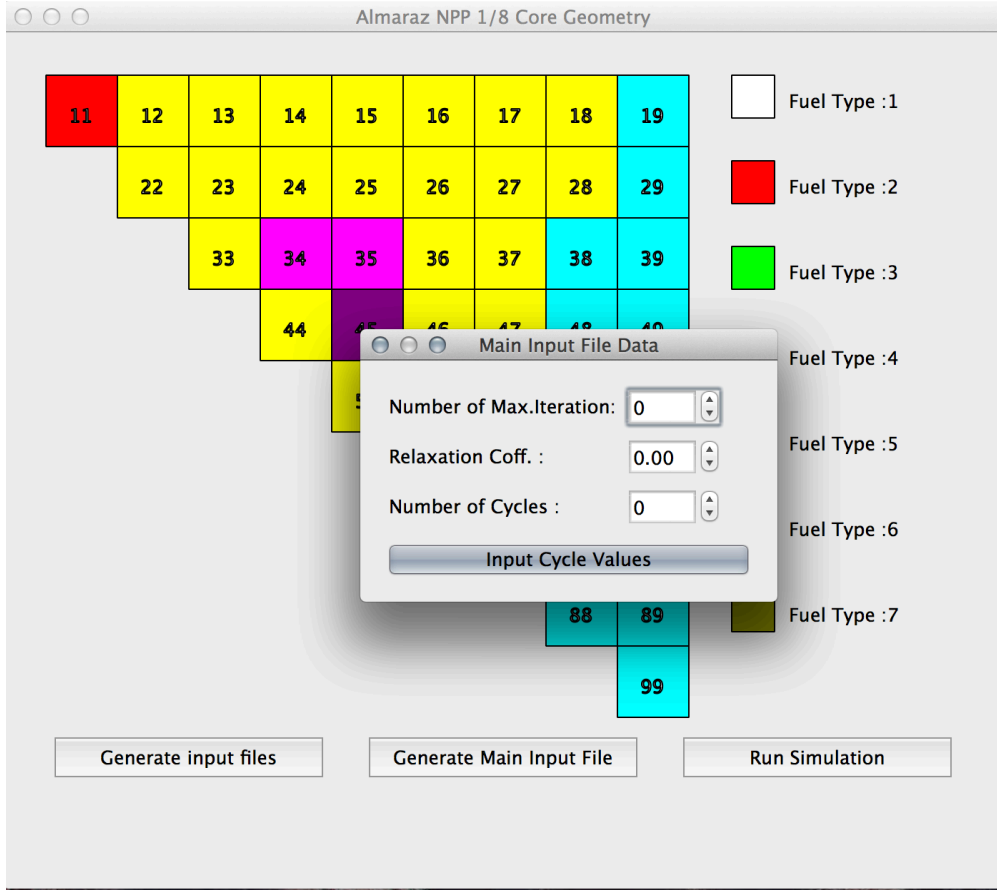
Şekil 6.7 Yakıt demetlerinden tipi belirlenmiş olanların görüntüsü

Yakıt demeti için tip belirlemesi ve sızan reaktivite ile sızan yüzey sayısı belirlendikten sonra seçilen yakıt demetinin tipine bağlı olarak şekil 6.7'de görüldüğü gibi yakıt demetinin rengi seçilen tipin rengine döner. Aynı zamanda yapılan bu değişiklikler Şekil 6.8'de verilen çıktı terminalinden de takip edilebilir. Çıktı terminali yakıt demeti üzerinde yapılan işlemleri ayrıntılı olarak ekrana yazar. Kullanıcının yaptığı herhangi bir işlemi unutmaması veya takip etmek istemesi durumunda kolayca bu terminale ulaşarak işlemlerini ve sıralamasını görebilir.

```
Output Terminal
(0, '11')
*****assembly properties*****
*****assembly ID: 11 *****
Assembly type is: 2
Leakage reactivity of assembly: 0.0
Number of leaked surfaces: 0
Color changed
(18, '34')
*****assembly properties*****
*****assembly ID: 34 *****
Assembly type is: 4
Leakage reactivity of assembly: 0.0
Number of leaked surfaces: 0
Color changed
(19, '35')
*****assembly properties*****
*****assembly ID: 35 *****
Assembly type is: 4
Leakage reactivity of assembly: 0.0
Number of leaked surfaces: 0
Color changed
(25, '45')
*****assembly properties*****
*****assembly ID: 45 *****
Assembly type is: 6
Leakage reactivity of assembly: 0.0
Number of leaked surfaces: 0
Color changed
```

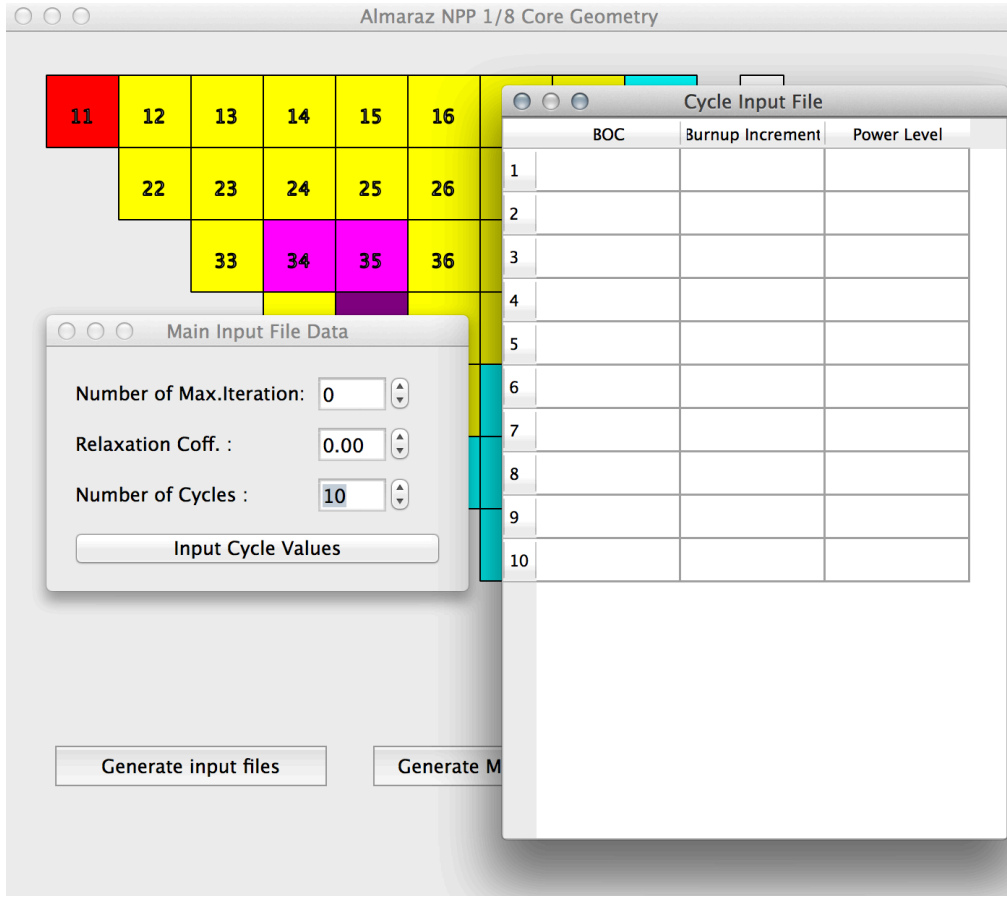
Şekil 6.8 PyRPM program çıktı terminali

Yakıt demeti tiplerinin seçimi ardından kullanıcı “Generate Input File”(girdi dosyasını oluştur) butonuna tıklayarak ilk girdi dosyasını oluşturur. Bunun ardından “Generate Main Input File” butonuna tıklayarak ana sistem için ana girdileri gelen ekrandan girmesi gerekir. Bu butona tıklamasıyla birlikte şekil 6.9’ da görülen “Main Input File Data” (ana girdi dosyası verileri) başlıklı pencere açılır. Bu pencerede sistemin hesaplama için yapacağı en fazla döngü sayısı(Number of Max. Iteration), hesaplama katsayısı (relaxation coefficient) ve sistemin kaç durum için hesaplayacağı(Number of Cycles) değerleri girilir.



Şekil 6.9 Program ana döngü ve iterasyon sayılarının belirlenmesi

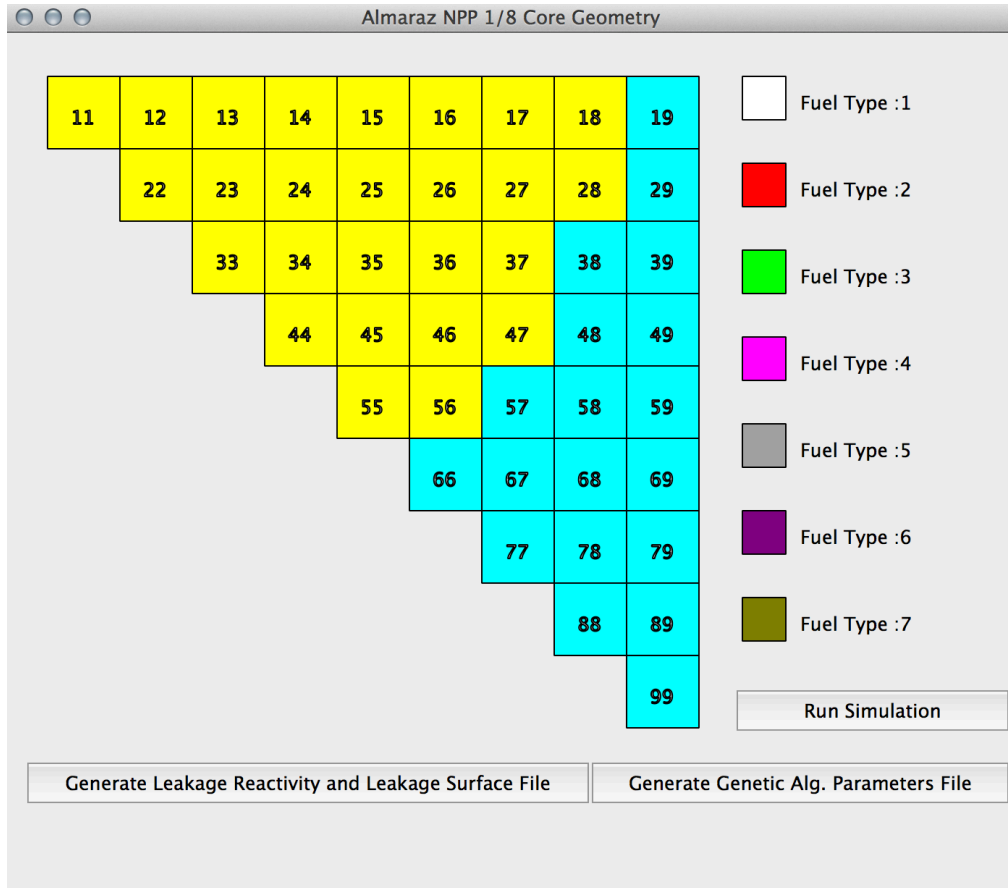
Number of Cycles değeri 0 dan büyük olmak zorundadır. Number of Cycles değerinin girilmesiyle birlikte atanan değer kadar hesaplama basamağı ve durumu oluşturulacağından her hesaplama için gerekli parametrelerin olduğu şekil 6.10'da ki ekran açılır. Bu ekrandaki BOC (Beginning of Cycle) yanma düzeni için girilen yakıt yükleme düzeninin başlangıç olup olmadığını belirlemek içindir. Başlangıç ise "1" değeri değil ise 0 değeri atanmalıdır. Burnup Increment(Yakıt Yanma Oranındaki Artış) değeridir. Power Level (Güç Seviyesi) is girilen yakıt yanma düzeni için sistemin güç seviyesinin yüzdece oranıdır. Bu değerlerin her bir durum için girilmesinin ardından pencerenin kapatılmasıyla ana girdi dosyası otomatik olarak oluşturulmuş olacaktır. Tüm bu parametrelerin girilmesinin ardından ana ekrandaki "Run Simulation"(Simülasyonu Başlat) butonuna tıklanarak oluşturulan girdi dosyalarının çözücü program olan ve FORTRAN programlama diliyle yazılıp derlenmiş olunan çözme modülüne sevki sağlanır. Çözme modülü direkt olarak ana sayfadan çağrılır ve çıktıları ana sayfa ile birlikte gelen çıktı terminaline yönlendirilir. Problemin başarılı olarak çözülmesiyle birlikte "rpm2dhunem.out" isimli çıktı dosyası program tarafından aynı dizinde yaratılır.



Şekil 6.10 PyRPM Program Döngü Değerlerinin Girilmesi Ekranı

6.2. Kor Yükleme Düzeni için RPM-GENETİK ve Grafikselle Kullanıcı Arayüzü(PyRPM-GENETIC)

Almaraz NGS'nin kor yükleme düzeninin önden tayin edilerek hesap yapılması yerine, genetik algoritmanın RPM programına uygulanmasıyla birlikte programın güç santrali için en iyi kor yakıt yükleme düzenini hesaplaması sağlanmıştır. PyRPM programından farklı olarak GA fonksiyonları ana çözücü modül olan RPM programına uygulanmış ve grafiksel arayüz buna göre yeniden tasarlanmıştır. Bu uygulamada da aynı reaktör için tasarlandığından giriş sayfaları benzerlik göstermektedir. PyRPM programından farklı olarak bu arayüz de giriş parametreleri GA'nın ve hesaplamaların sınırlamalarına yönelik değiştirilmiştir.



Şekil 6.11 PyRPM-Genetik Ana Sayfası Görüntüsü

Şekil 6.11'de programın çalıştırılmasıyla birlikte gelen ana sayfa görüntüsü bulunmaktadır. Bununla birlikte şekil 6.12'de görülen ana çıktı terminaline bu programda da yer verilmiştir.

Genetic Parameters

Selection Type: [dropdown]

Cross-Over Probability: 0.00 [spin]

Cross-over Variable: [dropdown]

Cross Type [dropdown]

Mutation Probability [dropdown]

Mutation Type: [dropdown]

Random Walk Step Parameters : 0 [spin] 0 [spin]

Numb. of Iteration: 0 [spin]

Numb. of Evaluation: 0 [spin]

Fitness Type: [dropdown]

Power Peaking Faktor 0.00 [spin]

Boron Cons.(ppm) 0 [spin]

Fitness weights

Burnup Set 0 [spin]

PPF Set 0.00 [spin]

Boron Set 0 [spin]

Weight Coefficients w1,w2,w3 0 [spin] 0 [spin] 0 [spin]

Set Genetic Parameters

Şekil 6.12 Genetik Parametreler Giriş Penceresi

Ana sayfanın açılmasıyla birlikte ekrana aynı zamanda genetik algoritma parametrelerinin de girileceği şekil 6.12 'de ki sayfa görünür. Bu penceredeki parametreleri kısaca tanımlamak gerekirse,

Selection Type(Seleksiyon Tipi): Oluşturulacak kromozomlar için seçim tipi belirlenir

Cross-Over Probability: Çaprazlama İhtimali atanır 0.0 ile 1.0 arasında bir değer olmalıdır.

Cross-Over Variable(Çaprazlama ihtimali değişkeni türü): Seçilen ihtimalin sabit veya rasgele olması ataması yapılır.

Mutation Probability (Mutasyon İhtimali Tipi) : Burada mutasyon tipinin sabit veya bir değere bağlı olması seçilir.

Mutation Type (Mutasyon Tipi): Mutasyonun bir veya iki noktada olacağı seçimi yapılır.

Random Walk Step Parameter (Rasgele Yürüyüş Basamak Parametreleri): Kromozom oluşturulurken hangi yakıt demetleri üzerine kaç basamakla ilerleyeceğinin değerleri girilir. Girilen değerler tam sayı olmak zorundadır.

Numb. of Iteration(Toplam İterasyon Sayısı): Programın en fazla kaç iterasyon yapacağı girilir. Girdi tam sayı olmak zorundadır.

Numb. of Evaluation (Toplam Evrim Sayısı): Oluşturulan kromozomlardan kaç evrim sağlanacağı girilir. Girdi tam sayı olmak zorundadır.

Fitness Type (Optimum Uygunluk Tipi): Hesaplanacak sonuç için en iyi sonucun kullanılacağı parametre seçilir.

Power Peaking Faktör (Güç En Yüksek Değeri): Hesaplamalarda tüm sistem için esas alınacak ve aşılmayacak en yüksek güç değeri.

Boron Cons.(ppm) (Boron Konsantrasyonu (milyonda bir parçacık sayısı)) : Hesaplamalarda tüm sistem için esas alınacak borik-asit miktarı.Girdi tam sayı olmak zorundadır.

Fitness Weights(Optimum Uygunluk Katsayıları):

Burnup Set: Yanma oranının sabitleneceği değer

PPF Set: Power Peaking Değerinin sabitleneceği değer

Boron Set: Borik-asit miktarının sabitleneceği değer

Wight Coefficients w1,w2,w3 : Burnup,PPF ve Boron için ağırlıklı katsayılarıdır.

Assembly Configuration

Assembly Size : 9 X 9

Number of Fuel Type: 7 Fuel Types

Population Size 0

Numb. of Popul. Size 0

Tot. Numb. of Fuel Type 1 0

Tot. Numb. of Fuel Type 2 0

Tot. Numb. of Fuel Type 3 0

Tot. Numb. of Fuel Type 4 0

Tot. Numb. of Fuel Type 5 0

Tot. Numb. of Fuel Type 6 0

Tot. Numb. of Fuel Type 7 0

Set Assembly Configuration

Şekil 6.13 Yakıt Demetleri Parametreleri Giriş Penceresi

Programın çalıştırılmasıyla açılan pencerelerden bir diğeri de yakıt demetlerinin özelliklerinin girileceği şekil 6.13 de görülen “Assembly Configuration” başlıklı penceredir. Buradaki buton ve başlık ve girdileri tanımlamak gerekirse,

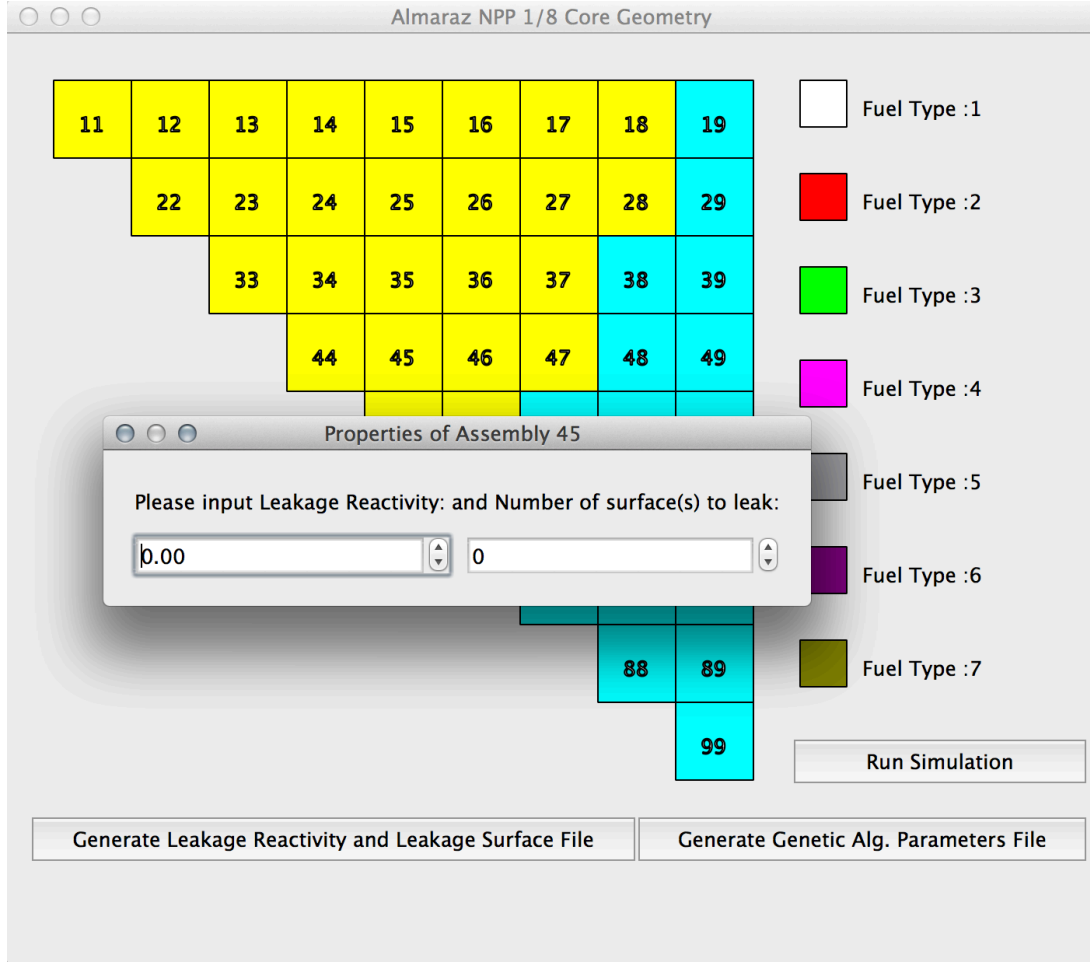
Assembly Size (Yakıt Demeti Boyutu): Bu değere readonly(sadece okunabilir) olarak ön tanımlı olarak belirlenmiştir. Bunun sebebi Almaraz NGS reaktör kore boyutunun tanımlı olmasıdır.

Number of Fuel Type(Toplam yakıt tipi): Almaraz NGS için ön tanımlı olarak yedi(7) yakıt demeti tipi belirlenmiştir. Fakat bu yakıt tiplerin tamamının kullanılma zorunluluğu yoktur.

Population Size (Popülasyon Sayısı): Genetik algoritma için kaç farklı popülasyonun kaç elemanlı olacağı girilir. Girdi tam sayı olmak zorundadır.

Numb. of Popul. Size (Toplam popülasyon tipi sayısı): Toplamda farklı kaç tip popülasyon oluşturulacağı girilir. Girdi tam sayı olmak zorundadır.

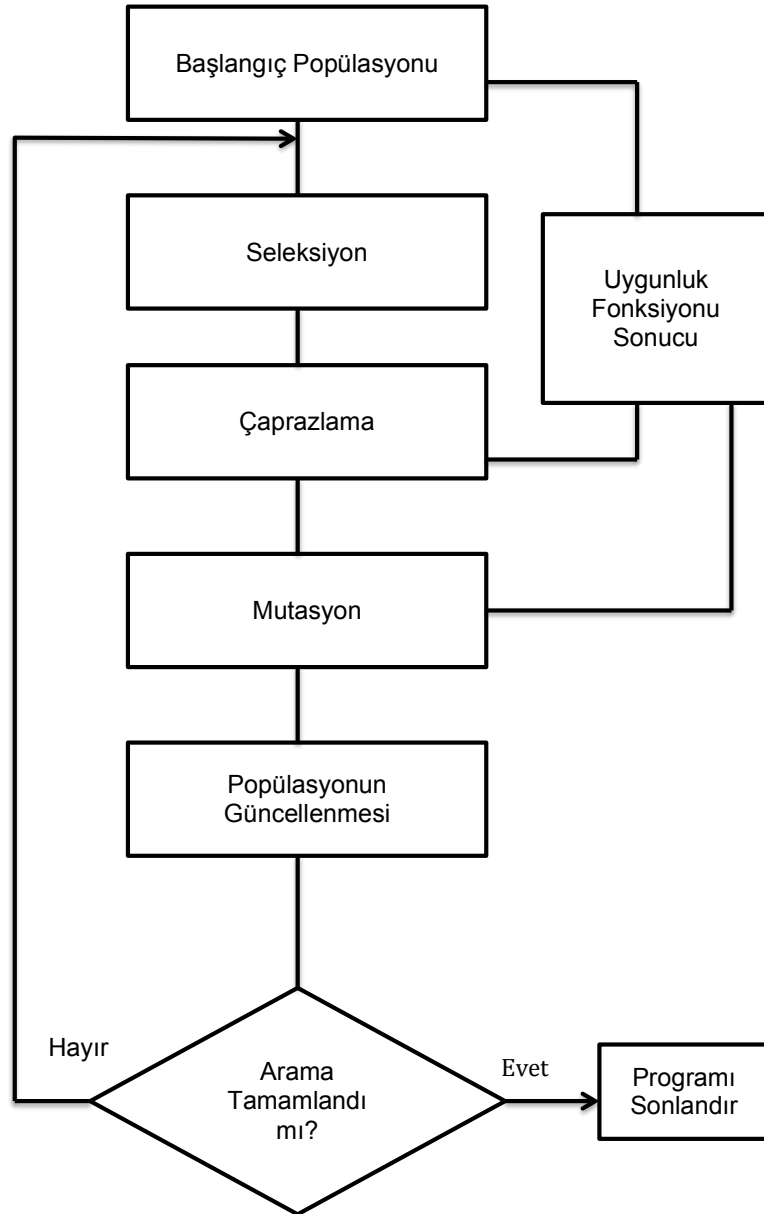
Tot. Numb. of Fuel Type X (Toplam Yakıt Demetinden Kaç adet olduğu): Reaktör kuru içerisinde toplamda X olarak belirtilen yakıt demeti tipinden kaç adet olduğu girilir. Girdi tam sayı olmak zorundadır.



Şekil 6.14 Yakıt Demetleri Sızıntı Yüzey ve Reaktivite Değerleri Giriş Penceresi

Yakıt tiplerinin ve sayılarının girişi yapıldıktan sonra ana pencere üzerinden şekil 6.14' de görüldüğü gibi yakıt demetine tıklanarak yakıt demetinden sızan reaktivite ve yüzey sayıları girilir. Yüzey sayıları sadece açık mavi bölgeye komşuluğu olan yakıt demetleri içindir. Sızan reaktivite ve yüzey sayıları değerlerinin girilmesinin müteakip "Generate Leakage Reactivity and Leakage Surface File" butonuna basılarak değerlerin girdi dosyasına yazılması sağlanır.

Tüm yakıt demetlerinin değerlerinin girilmesinin ve genetik parametrelerin ayarlanmasından sonra “Generate Genetic Alg. Parameters File” butonuna basılarak yakıt demeti verilerinin ve genetik algoritma verilerinin bulunduğu dosya yaratılır. Ana pencere üzerinde bulunan “Run Simulation” butonuna basılarak girdi dosyalarının önceden oluşturulmuş ve derlenmiş olan RPM-Genetic programına geçirilmesi sağlanır.



Şekil 6.15 Genetik Algoritma Akı Şeması [18]

RPM-Genetic programı Şekil 6.15'de gösterilen akış şemasındaki optimizasyon yöntemi ile RPM programı üzerinden işlemi gerçekleştirecektir.

Bu süreç sırasındaki tüm girdi ve çıktı işlemleri çıktı terminalinden takip edilir. Programın sonuçlanmasıyla birlikte girilen parametrelerin değerlerinin bulunduğu "norm-" kelimesi ile başlayan bir adet data bir adet out uzantılı iki dosya oluşur. ".out " uzantılı dosya girilen verilere göre en iyi yakıt yükleme düzenini ve hesaplanan değerlerin kısa bir özetini gösterir. ".dat" uzantılı çıktı dosyası ise her hesaplama için bulunan tüm değerlerin olduğu dosyadır. Her iki dosyada text formatında olup herhangi bir editör ile açılabilir.

7. Örnek Problemlerin Uygulanması

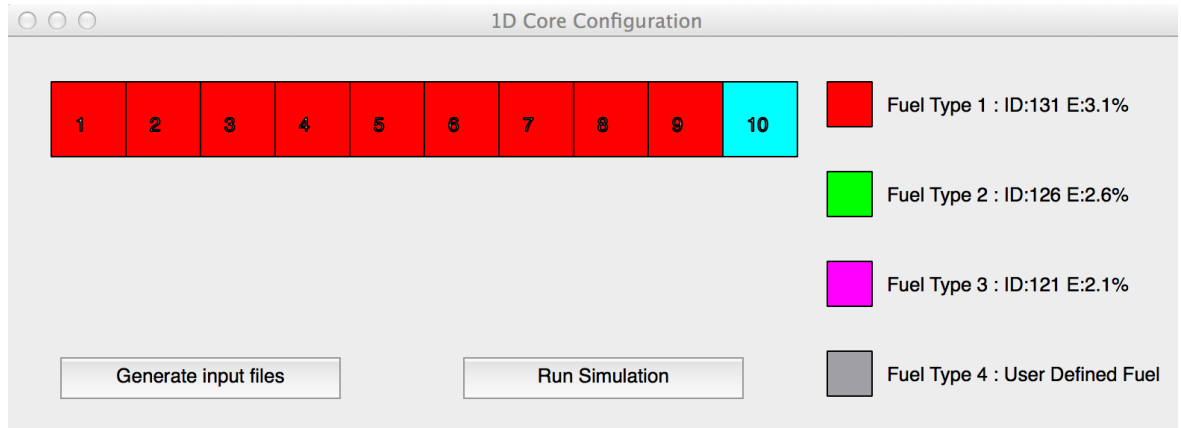
7.1. Tek Boyutlu Kor Yükleme Düzeni İçin Çeşitli Uygulamalar

Herbir program için model problemleri ve programları incelemek gerekirse tek boyutlu reaktör kor modelini anlamak tüm problemler için bir temel teşkil edecektir. Tek boyutlu kor modelinde üç farklı zenginlikte basınçlı su reaktörü yakıtı kullanılmıştır. Çözüm işlemini gerçekleştirecek olan ana FORTRAN programına bu yakıtlarla ilgili katsayılar önceden tanımlanmıştır. Kullanıcı bu yakıt tiplerini reaktör içerisinde istediği sıra ve yapı ile dizerek yanma oranlarını, boron miktarını ve herbir yakıttaki güç oranını elde etmesi hedeflenmiştir.

Yakıt tiplerini listelemek gerekirse:

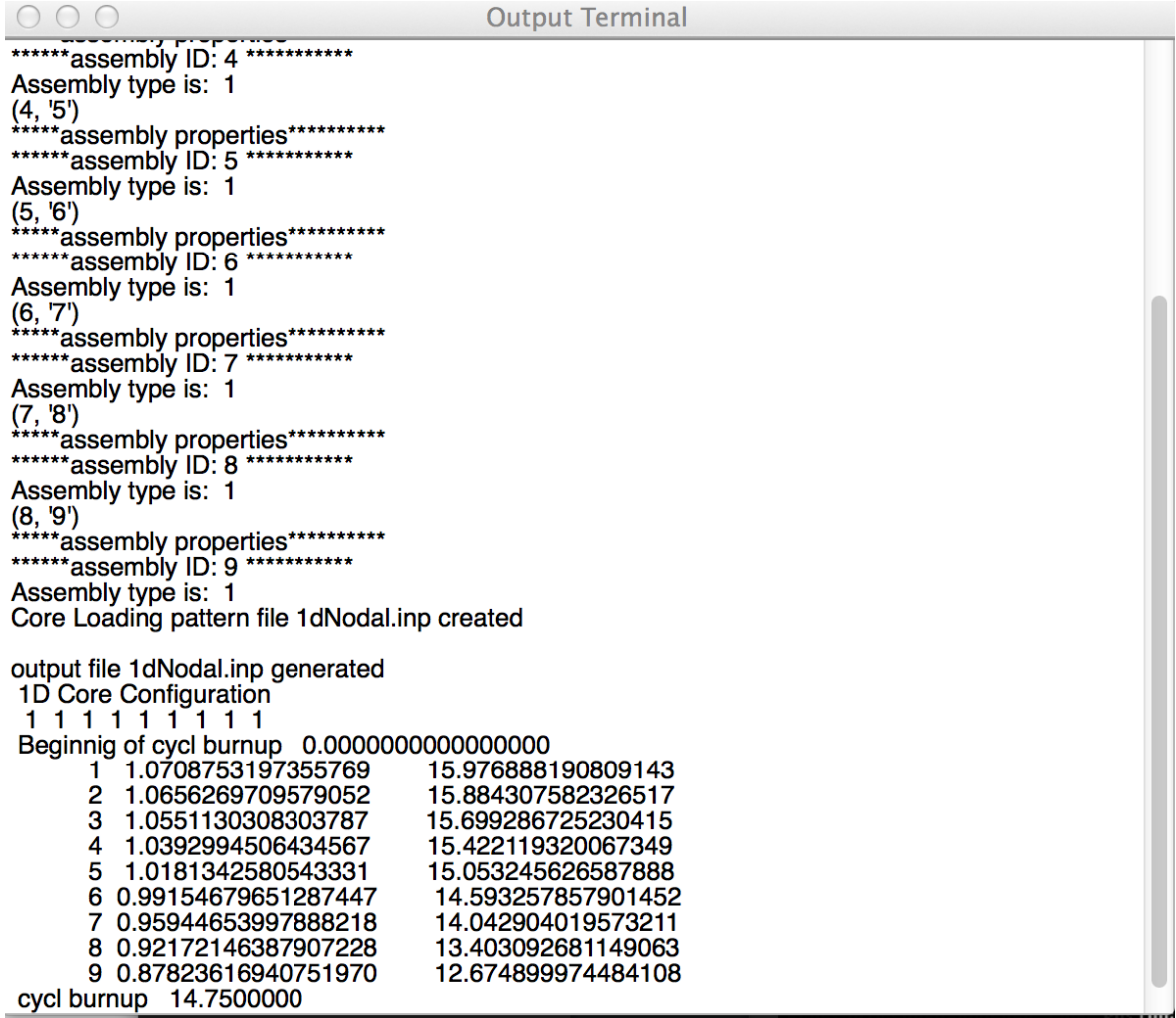
1. Tip Yakıt %3.1 zenginlikte
2. Tip Yakıt %2.6 zenginlikte
3. Tip Yakıt %2.1 zenginliktedir.
4. Tip Yakıt Kullanıcının Özelliklerini Tanımlayabileceği Yakıt(kaynak programlarda değişikliğe gidilmesi şartı ile)

Bu yakıt tipleri ile ilk Py1DNodal programını çeşitli yakıt yükleme düzenleri için örnekleme gerekirse:



Şekil 7.1 Py1DNodal Giriş Arayüzü

Şekil 7.1’de görülen program arayüzü ile bir boyutlu kor 1. Tip yakıt seçilerek düzenlenir ardından girdi dosyası (Generate Input Files) butonu oluşturularak ilk problem için “Run Simulation” butonu ile çalıştırılırsa çıktı terminalinde aşağıdaki sonuç oluşacaktır.

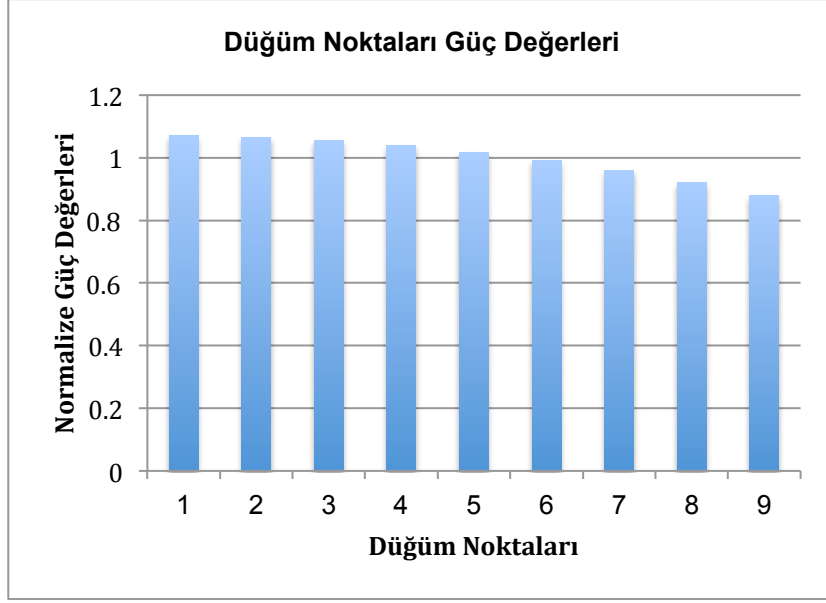


```
*****assembly ID: 4 *****
Assembly type is: 1
(4, '5')
*****assembly properties*****
*****assembly ID: 5 *****
Assembly type is: 1
(5, '6')
*****assembly properties*****
*****assembly ID: 6 *****
Assembly type is: 1
(6, '7')
*****assembly properties*****
*****assembly ID: 7 *****
Assembly type is: 1
(7, '8')
*****assembly properties*****
*****assembly ID: 8 *****
Assembly type is: 1
(8, '9')
*****assembly properties*****
*****assembly ID: 9 *****
Assembly type is: 1
Core Loading pattern file 1dNodal.inp created

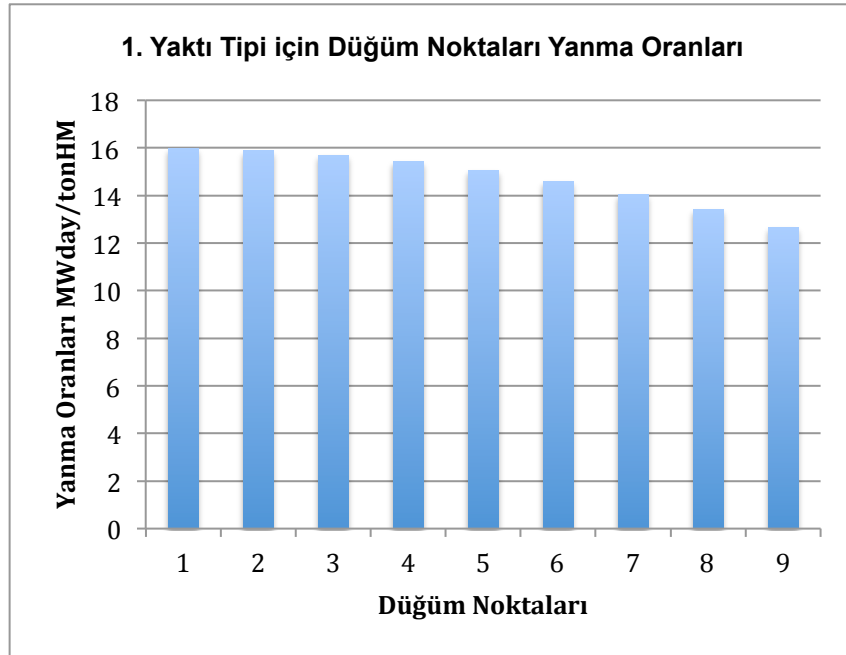
output file 1dNodal.inp generated
1D Core Configuration
1 1 1 1 1 1 1 1 1
Beginnig of cycl burnup 0.0000000000000000
  1 1.0708753197355769    15.976888190809143
  2 1.0656269709579052    15.884307582326517
  3 1.0551130308303787    15.699286725230415
  4 1.0392994506434567    15.422119320067349
  5 1.0181342580543331    15.053245626587888
  6 0.99154679651287447    14.593257857901452
  7 0.95944653997888218    14.042904019573211
  8 0.92172146387907228    13.403092681149063
  9 0.87823616940751970    12.674899974484108
cycl burnup 14.7500000
```

Şekil 7.2 Py1DNodal Çıktı Terminali

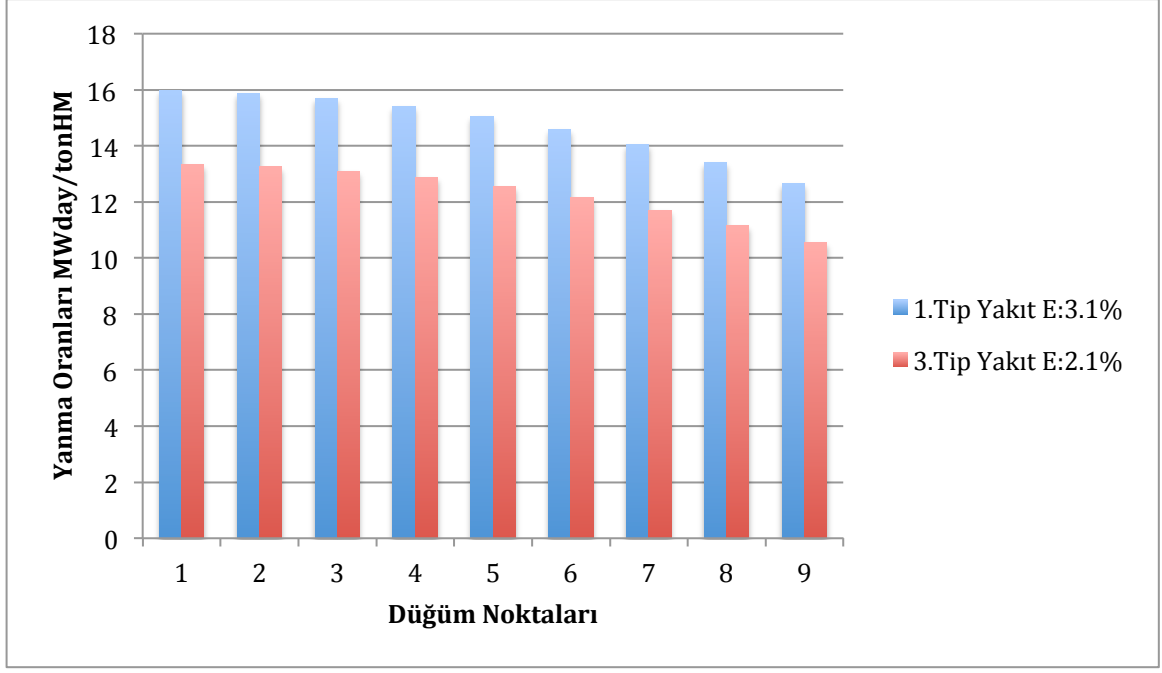
Programın sonuçları çıktı dosyaları şeklinde de ayrıca incelenebilir. Çıktı dosyaları kullanılarak noktasal güç değerlerini ve yanma oranlarını grafiksel olarak göstermek gerekirse:



Şekil 7.3 1DNodal 1.Yakıt Tipi için Normalize Güç Değerleri



Şekil 7.4 1DNodal 1.Yakıt Tipi için Yanma Oranları



Şekil 7.5 1DNodal 1. Ve 3. Tip Yakıtların Yanma Oranları

Tüm grafikler incelendiğinde yanma eğrileri ile güç eğrilerinin uyumlu olduğu ve daha zengin olan yakıtın daha yüksek yanma oranlarına ulaştığı görülmektedir. Eğrilerin ayrıca levha geometrisi için bir-birbuçuk difüzyon denkleminin çözümünü nümerik olarak sağladığı görülmektedir. Ayrıca program çıktı dosyaları üzerinden tüm değerler ayrıntılı olarak herhangi bir metin editörü ile kolaylıkla okunabilecek şekilde tasarlanmıştır.

7.2. Almaraz NGS İki Boyutlu Kor Yükleme Düzeni İçin Çeşitli Uygulamalar

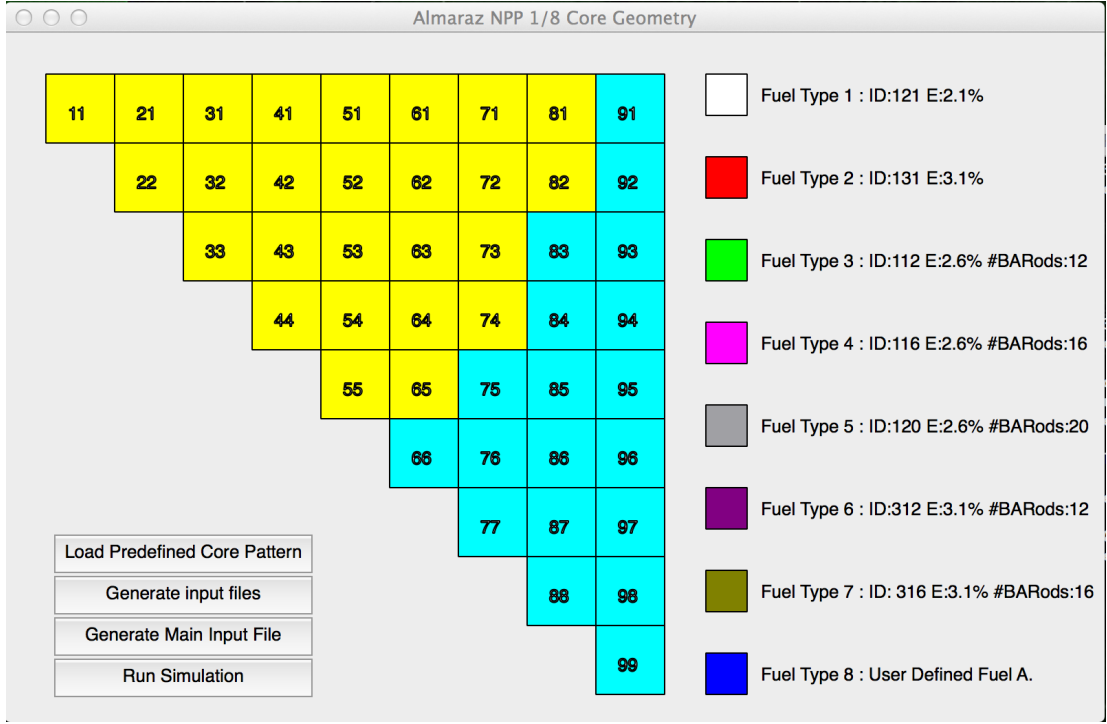
Bir önceki problemin temel alt yapısı kullanılarak geliştirilmiş olan RPM(reloading power mapping) programı Almaraz NGS için bir yakıt yükleme simülasyonu niteliğindedir. Bu amaçla aşağıda tipleri verilmiş olan yakıt demetleri kullanılarak iki boyutlu bir kor düzeni modellemesi gerçekleştirilebilir. Geliştirilmiş olunan PyRPM arayüzü ile yakıt düzeni ve reaktördeki sızıntı reaktiviteleri tanımlanarak verilen yanma değeri için gerekli boron miktarı, yakıt demeti normalize güç oranları ve yanma oranları programca hesaplanarak çıktı dosyaları oluşturulur.

Yakıt Demetleri ve Özellikleri:

Çizelge 7.1 Almaraz NGS yakıt tipleri

Tipi	Etiketi:	Yakıt Zenginliği (%)	Yanıcı Zehirli Çubuk Sayısı
M-20: (5)	ID:120	2.6	20
M-16: (4)	ID:116	2.6	16
M-12: (3)	ID:112	2.6	12
N-16: (7)	ID:316	3.1	16
N-12: (6)	ID:312	3.1	12
L: (1)	ID:121	2.1	0
N-0: (2)	ID:131	3.1	0
8	Kullanıcı tarafından tanımlanacak yakıt demeti		

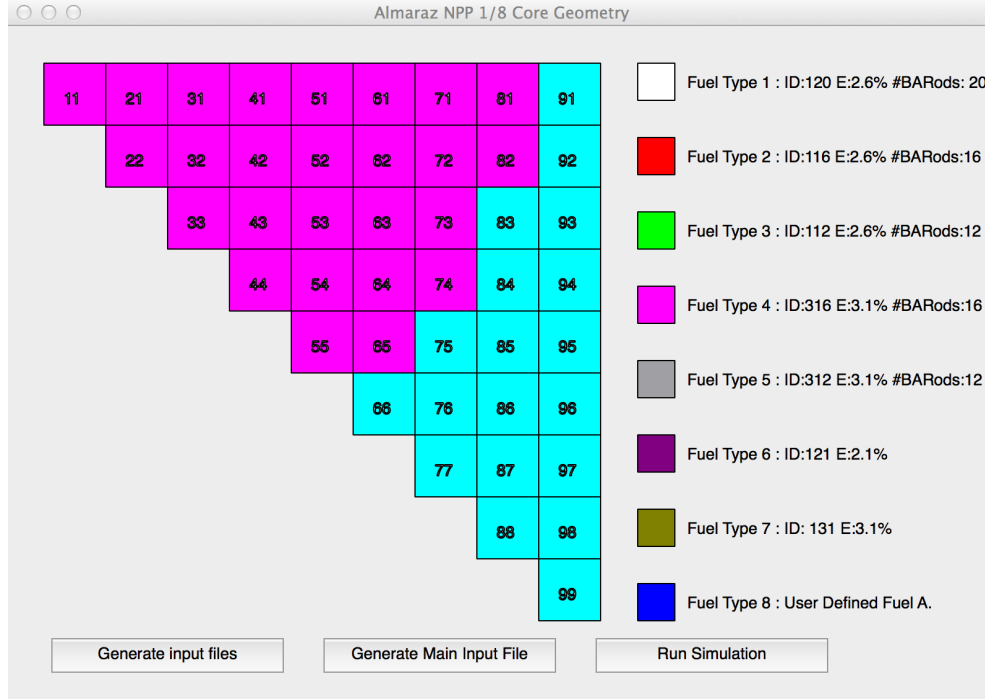
Tüm reaktör korunu tek tip yakıt ile doldurarak yanıcı zehirli çubukların etkilerini, yanma öncesi ve sonrası normalize güç değerlerini, yanma oranlarını hesaplayacak şekilde programımızı arayüz ile düzenlersek sırasıyla aşağıdaki basamakları takip etmemiz gerekecektir.



Şekil 7.6 PyRPM Giriş Penceresi

Tüm yakıt tipleri her seferinde yalnızca tek tip yakıt demeti kullanılmak şartı ile aşağıdaki gibi düzenlenirse;

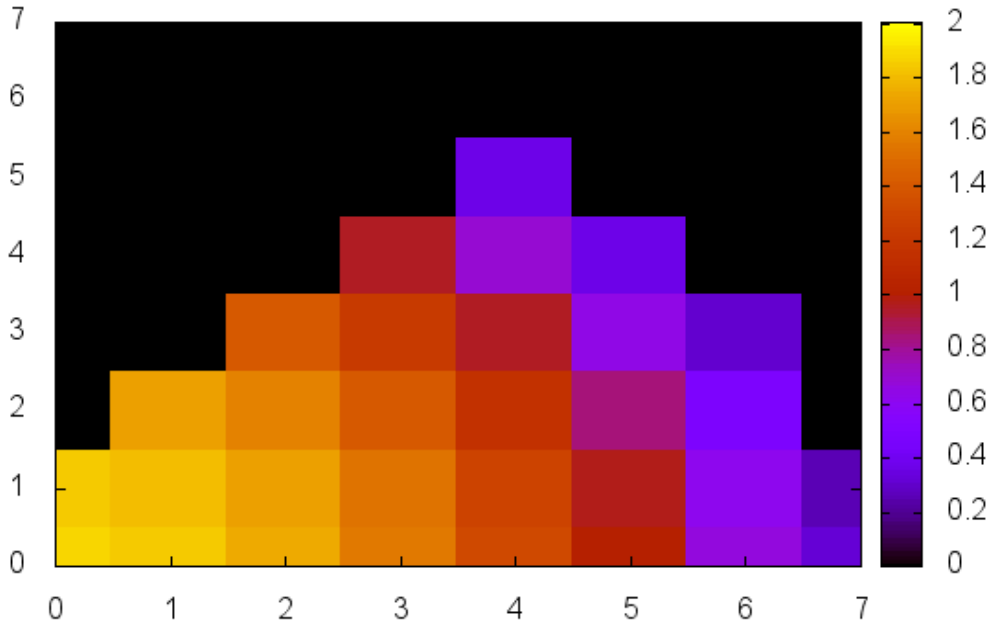
İlk adımda tüm kor yapısını 4. Yakıt tipi ile doldurursak:



Şekil 7.7 PyRPM Giriş Penceresinde Yakıtların Düzenlenmiş Görüntüsü

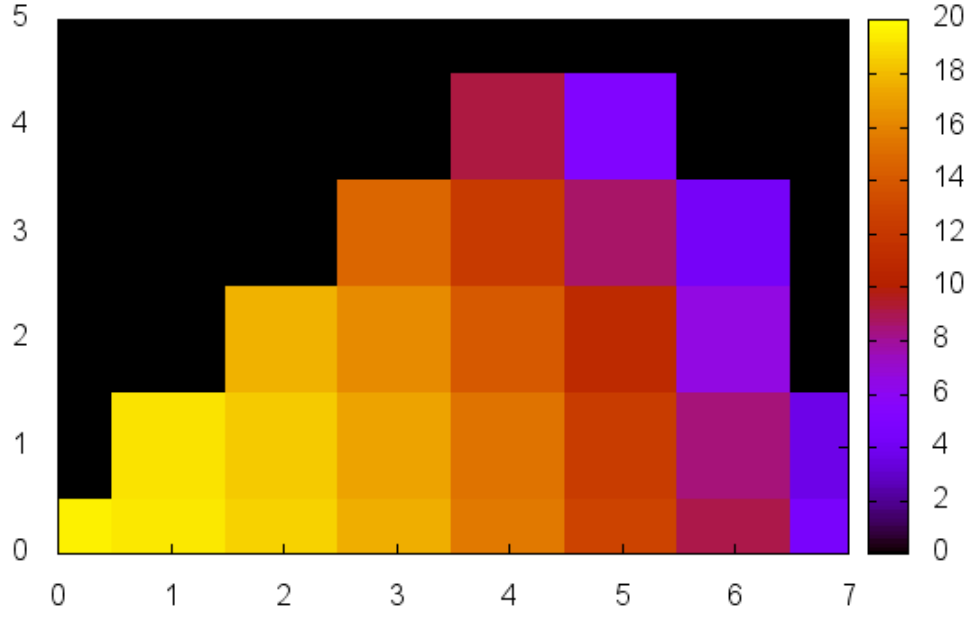
Şekil 7.6'da görülen arayüz üzerinden tüm yakıt demetleri 4. Tip seçilir ve 65,74,73,82,81 numaralı yakıt demetleri için sızıntı reaktivitesi ve sızıntı yüzeyleri sayısı tanımlanıp "Generate Input Files" butonuna tıklanarak ilk girdi dosyası oluşturulur. Ardından ortalama yanma miktarını, döngü başlangıcını(1 veya 0 olacak şekilde) ve güç seviyesini tanımlamak için "Generate Main Input File" butonuna tıklanarak gerekli bilgiler girilir. Bu örnek için tam güçte olan bir sistemin 12000 MWday/ton yanma oranı ile ilk döngüye girecek şekilde tanımlanmıştır. Ardından işlemi başlatmak için "Run Simulation" butonuna tıklanarak hesaplama başlatılır ve sonuç dosyaları aynı klasörde oluşur. Ayrıca çıktı terminali üzerinden de sonuçlar takip edilir. Aynı adımlar diğer Yakıt tipleri içinde tekrarlanıp çıktı dosyaları elde edildikten sonra tüm durumları kıyaslamak gerekirse;

Güç profili tüm yakıt tipleri için aynı olacağından aşağıdaki gibi olacaktır.



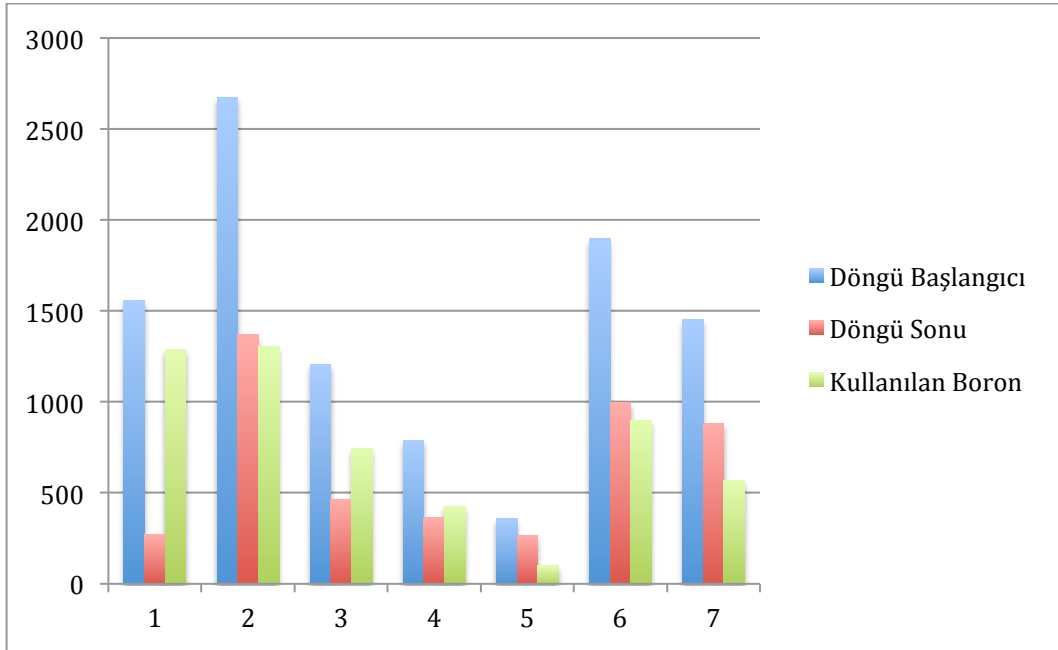
Şekil 7.8 Normalize Güç Oranları

Yakıt demetleri için yanma miktarları profili de güç profiline benzer şekilde aşağıdaki gibi olacaktır.



Şekil 7.9 Yakıt Demetleri Yanma Oranları

Tüm yakıt demetleri için en büyük farklılık aynı yanma oranı için döngü sonundaki boron konsantrasyonu miktarıdır. Bu farkı grafikte göstermek gerekirse, yakıt zenginliğinin reaktivite üzerindeki etkisi kolayca görülecektir.



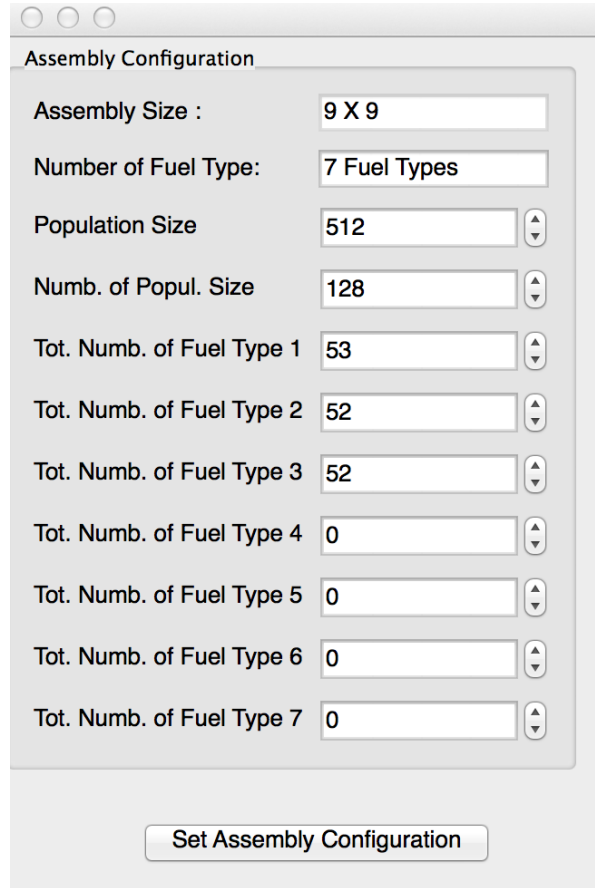
Şekil 7.10 Boron Konsantrasyonu Değişimi

PyRPM programı kullanılarak Almaraz NGS için çeşitli yakıt yükleme modelleri iki boyutlu olarak düzenlenmiş ve aynı yanma oranı için güç profili, yanma oranı profili ve boron konsantrasyonu profilleri elde edilmiştir.

7.3. Almaraz NGS Kor Yükleme Düzeni için Genetik Algoritma Uygulaması

Genetik algoritma yaklaşımı bir problemin çeşitli şartları sağlamak koşulu ile en optimize sonucu verecek şekilde tasarlanmıştır. Almaraz NGS için önceden tanımlı şartları sağlamak koşulu ile olabilecek en iyi yakıt düzenini hesaplayacak şekilde tasarlanmıştır. Aşağıda belirlenmiş şartlar kullanılarak örnek bir kor yükleme düzenini hesaplamak için aşağıdaki adımları takip edebiliriz. PyRPMGenetic programı için gerekli girdi parametrelerini program arayüzü üzerinden açılan temel üç pencere ile yapılmaktadır.

1. Yakıt Demeti Penceresi ve GA Popülasyon Özellikleri :



Parameter	Value
Assembly Size	9 X 9
Number of Fuel Type	7 Fuel Types
Population Size	512
Numb. of Popul. Size	128
Tot. Numb. of Fuel Type 1	53
Tot. Numb. of Fuel Type 2	52
Tot. Numb. of Fuel Type 3	52
Tot. Numb. of Fuel Type 4	0
Tot. Numb. of Fuel Type 5	0
Tot. Numb. of Fuel Type 6	0
Tot. Numb. of Fuel Type 7	0

Set Assembly Configuration

Şekil 7.11 Yakıt Demetleri ve Popülasyon Sayısı Penceresi

Program Almaraz NGS için tasarlandığından seçilecek yakıt demeti tipinin sayısının toplamda 157 olması gerekmektedir. Ayrıca popülasyon büyüklüğü ve adeti kullanıcı tarafından belirlenebilir.

2. Genetik Özellikler ve Sınır Koşulları Penceresi

Genetic Parameters

Selection Type: Two individuals with group probability

Cross-Over Probability: 0.80

Cross-over Variable: Fix Cross-Over

Cross Type: Two-Point Cross-Over

Mutation Probability: Fix Probability

Mutation Type: One-Point mutation

Random Walk Step Parameters : 17 3

Numb. of Iteration: 500

Numb. of Evaluation: 200

Fitness Type: With Weight Coff.

Power Peaking Faktor: 1.44

Boron Cons.(ppm): 1200

Fitness weights

Burnup Set: 2

PPF Set: 2.70

Boron Set: 1000

Weight Coefficients w1,w2,w3: 2 120 1

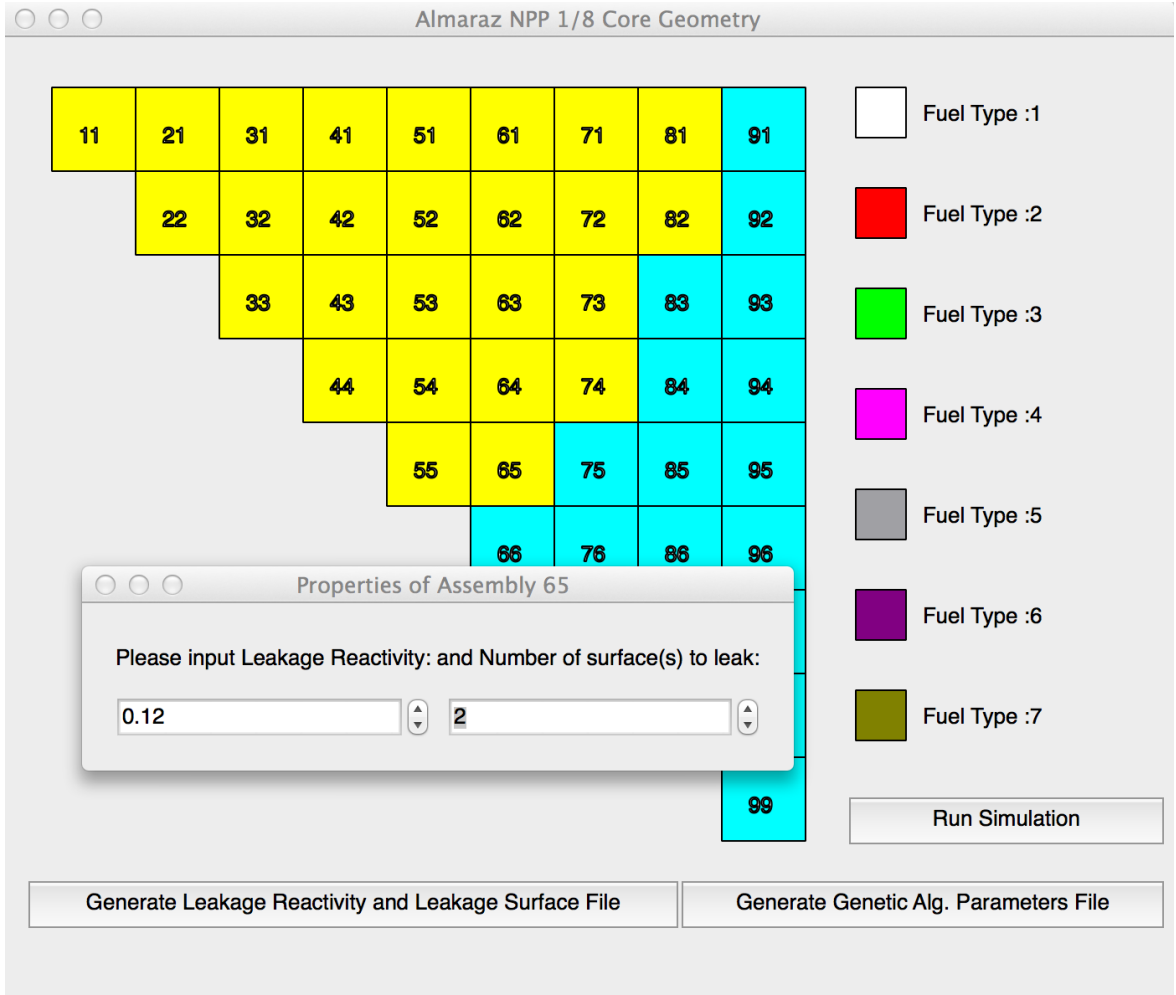
Set Genetic Parameters

Şekil 7.12 Genetik Parametreler Giriş Penceresi

Genetik parametlerin ve sınır şartları yukarıdaki şekilde gösterildiği gibi tanımlandıktan sonra “Set Genetic Parameters” butonu ile gerekli girdi dosyası oluşturulur.

Son olarak ana pencere ile kor düzenindeki sızıntı reaktiviteleri olan yakıt demetleri ve sızıntı yüzey sayıları tanımlanır.

3. Ana Pencere



Şekil 7.13 PyRPMGenetic Ana Pencere

Gerekli tanımlamalar yapıldıktan sonra “Generate Leakage Reactivity and Leakage Surface File” butonuna tıklanarak sızıntı reaktivite bilgisini içeren girdi dosyası oluşturulur ardından “Generate Genetic Alg. Parameters File” butonuna tıklanarak GA bilgilerini içeren girdi dosyası oluşturulur. “Run Simulation” butonuna tıklanarak işlem başlatılır.

İşlemin tamamlanmasının ardından program tarafından iki adet çıktı dosyası üretilir. Bu dosyalar *.dat uzantılı olan; program boyunca hesaplanan verileri herhangi bir yorum satırı içermeden başka bir programda girdi dosyası olarak kullanılabilir formattadır.

Diğer çıktı dosyası ise *.out uzantılı olup programın girdi bilgilerini ve çıktı bilgilerini bir özet olarak verir. Yukarıdaki problem için out dosyasını bir bölümü örnek olarak aşağıda verilmiştir.

I N P U T

Pattern size 9
 Number of enrichment 7
 Number of population size 512

Your pattern looks like (X=Fuel,0=Water):
 1 2 3 4 5 6 7 8 9

```

X X X X X X X X 0 1
X X X X X X X X 0 2
X X X X X 0 0 3
X X X X 0 0 4
X X 0 0 0 5
0 0 0 0 6
0 0 0 7
0 0 8
0 9
    
```

Assembly Type	Total Number in Full Core
1	53
2	52
3	52
4	0
5	0
6	0
7	0

Selection type is -- Two individuals according to their group probability --

Cross-Over Probability 0.8
 Mutation Probability 0.2

Mutation type is -- One-point Mutation --

Random walk step number is various between 3 and 20

Number of iteration 500

Fitness depends on -- Weighted Fitness Function-- where;

Setted Burnup 2.0 and its weight 2.0
 Setted ppf 2.7 and its weight 120.0
 Setted Boron Concentration 1000.0 and its weight 1.0

popSZ	# of iteration	# of evaluation	best fitness	BOC ppf	EOC ppf	Cycle Burnup	Boron Cont.	Ave. BOCppf	Ave. EOCppf	Ave. Burnup	var. fitness	var. BOCppf	var. EOCppf	var. Burnup
512	8	3672	799.66	1.421	1.429	16.736	229.82	1.483	1.375	17.12	0.504	0.0057	0.0029	0.0105

Best Pattern

```

5 5 7 4 5 5 4 7
5 5 4 5 4 7 7
5 5 5 4 4
4 4 7 7
7 7
    
```

cpu time for rpm : 5.22
 cpu time for genetic : 0.48
 total cpu time for this iteration : 5.70

Şekil 7.14 Çıktı Dosyası Sonucu

8. SONUÇLAR

Bu çalışmada kor içi yakıt yükleme analizinde kullanılmak amacıyla bir boyutta ve iki boyutta basitleştirilmiş yazılımlar arayüzler de yazılarak kullanılmak üzere geliştirilmiştir. Almaraz NGS'nin test yazılımlar için sunulan değerleri referans alınarak, reaktör için en iyi yakıt yükleme düzenini hesaplayacak bilgisayar programı içinde arayüz yazılmıştır. İki boyutlu hesaplamalarda Almaraz NGS'nin değerleri esas alınmıştır bu sebeple programın herhangi başka bir nükleer reaktör için veya başka özellikler taşıyan yakıt demetleri girdi tanımlarında değişiklikler yapılması gerekmektedir.. Yazılan bilgisayar programları son geliştirici ve son kullanıcıyı hedeflediğinden programın parçalarını problemlerle birlikte kısaca özetlenmektedir.

Bir boyutta yanma oranına bağlı hesaplamalar için 1DNodal yöntemler kullanan program geliştirilmiştir, iki boyutta ise Almaraz NGS IAEA-TECDOC-815' te verilen değerleri baz alarak için en iyi kor yükleme düzeninin hesaplanması için bilgisayar programı grafiksel kullanıcı arayüz paketiyle hazırlanmıştır.

İlk grafiksel kullanıcı arayüz programı 1DNodal modülüne girdi hazırlamak için tasarlanmıştır. İkinci arayüz ise iki boyutlu geometrideki RPM modülüne girdi dosyaları hazırlamak için tasarlanmıştır. RPM modülü çalışma sonucunda reaktör için nükleer yakıt demetlerinin yanma oranlarına karşılık güç değerlerini hesaplayarak çıktı olarak vermektedir. Çıktı dosyası aşağıdaki şekilde olmaktadır.

```

rpm2dhunem.out
1 2 2 1 1 4 6 5 0
0 3 3 6 3 6 4 4 0
0 0 5 7 2 5 3 0 0
0 0 0 7 1 2 5 0 0
0 0 0 0 2 7 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
burnup= 0.000000 2.277605
2.0705 2.2776 1.9300 1.2384 0.9007 0.6051 0.4648 0.2545
2.2776 1.6459 1.3331 1.2337 0.9430 0.7313 0.4299 0.2645
0.0000 1.3331 0.9916 1.1174 1.2655 0.7415 0.5428 0.0000
0.0000 0.0000 1.1174 1.1356 1.1918 1.2061 0.5909 0.0000
0.0000 0.0000 0.0000 1.1918 1.5572 1.2742 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 1.2742 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
1930.249 ppm
iteration number
39
burnup= 1.250000E-02 2.020478
1.8553 2.0205 1.7586 1.1972 0.9193 0.6416 0.5167 0.2984
2.0205 1.4749 1.2316 1.1839 0.9530 0.7684 0.4761 0.3106
0.0000 1.2316 0.9473 1.1022 1.2942 0.7798 0.5928 0.0000
0.0000 0.0000 1.1022 1.1448 1.2365 1.2641 0.6367 0.0000
0.0000 0.0000 0.0000 1.2365 1.5949 1.3062 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 1.3062 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
1702.875 ppm
burnup each ass
0.02 0.03 0.02 0.02 0.01 0.01 0.01 0.00
0.00 0.02 0.02 0.02 0.01 0.01 0.01 0.00
0.00 0.00 0.01 0.01 0.02 0.01 0.01 0.00
0.00 0.00 0.00 0.01 0.02 0.02 0.01 0.00
0.00 0.00 0.00 0.00 0.02 0.02 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

```

Şekil 8.1 PyRPM Program Çıktı Sayfası

Sonraki grafiksel kullanıcı arayüz programı ise RPM-Genetic modülüne girdi hazırlamak amacıyla tasarlanmış, girdilerin tamamlanıp programın çalıştırılmasıyla birlikte Almaraz NGS'i için en iyi kor yüklenme düzeni girdilere uygun olarak genetik algoritma kullanılarak hesaplanmıştır. Bu program için örnek çıktı dosyaları aşağıdaki gibidir.

```

O U T P U T S
-----
# of # of best BOC EOC Cycle Boron Ave. Ave. Ave. var. var. var. var.
popsz iteration evaluation fitness ppf pff Burnup Cont. BOCppf EOCppf Burnup fitness BOCppf EOCppf Burnup
128 0 0 29508.21 2.426 1.322 29.508 1714.53 1.899 1.226 26.97 48.145 0.0324 0.0070 0.0481

Best Pattern
1 3 2 4 4 1 4 2
6 2 2 1 5 4 4
2 1 1 1 1
6 2 1 4
4 7

cpu time for rpm : 4.37
cpu time for genetic : 0.00
total cpu time for this iteration : 4.37

```

Şekil 8.2 PyRPM-Genetic Program Çıktı Sayfası

Hazırlanan her iki arayüz programı ve hesaplama modülleri için girdiler sırasında tüm kontrollerin yapılmaması sebebiyle bazı hatalar verebilmektedir. Tüm durumlar için senaryolar oluşturulsa bile program dahilinde gözden kaçan veya eksik kalan bazı noktalar olasıdır. Bunun dışında programın sonuçları hesaplanan ve deneysel verilerle uyum içerisindedir.

Buradaki temel hedef kullanıcının doğrusal reaktivite modeline aşina olmasını sağlamak ve nükleer reaktör yakıt elemanın yanmasındaki etkileri görebilmesini sağlamaktır. Grafikselleştirilmiş kullanıcı arayüzleri kullanıcının girdi dosyalarındaki keskin satır sütun konumlarını düşünmeden hangi yakıt demeti üzerinde işlem yaptığını rahatça görerek ve bunları istediği zaman değiştirmesini sağlayarak daha hızlı sonuç almasını sağlamıştır. Grafikselleştirilmiş arayüzün geliştirilmesiyle birlikte bir reaktörün 1/8 geometrisinin nasıl görüldüğü, kullanıcının daha yaygın olarak kullanılan nükleer reaktörler için hesaplama kodlarını kullanmaya başladığında girdi dosyalarında ne görmesi gerektiği konusunda aşina olması hedeflenmiştir. Program geliştirilmeye devam edilebilmesi açısından açık uçlu(explicit) olarak yazılmıştır. Programın modülleri ve arayüz programı birbirinden bağımsız olarak da ayrıca geliştirilmeye açıktır.

Sonuç olarak bu tez çalışmasında geliştirilen arayüzler kullanılarak eğitim amaçlı olarak bir ve iki boyutta yanma oranına bağlı nötronik hesaplamalar yapılabilmektedir. Bu hesaplamalar genetik algoritma kullanılarak optimizasyon amaçlıda kullanılabilir. Lisans ve yüksek lisans seviyesinde reaktör analizi ve yakıt yönetimi derslerinde hesaplamalı araçlar olarak kullanılması mümkün olacaktır.

KAYNAKLAR

- [1] Report for European Commission, *Main Characteristics of Nuclear Power Plants in the European Union and Candidate Countries*, AEAT/R/PSEG/0405 Issue 3, **2001**.
- [2] Spanish Nuclear Power Plants in UNESA, **1999**.
- [3] Las Centrales Nucleares Españolas, 2a Edición, Consejo de Seguridad Nuclear.
- [4] FSAR Rev. AC11
- [5] IAEA, *In-Core Fuel Management Code Package Validation for PWRs*, IAEA-TEC-DOC-815, Vienna, **1995**
- [6] Fortran Programming Language, <http://groups.engin.umd.umich.edu/CIS/course.des/cis400/fortran/fortran.html>, Ağustos, **2013**).
- [7] Guido van Rossum, "Why was Python created in the first place?". General Python FAQ. Python Software Foundation. Retrieved 22 March **2007**.
- [8] Python About Page, <http://www.python.org/about/>, (Ağustos **2013**).
- [9] Qt-Project, <http://qt-project.org/>, (Ağustos, **2013**).
- [10] Bell, G.I., Glasstone, S., Nuclear Reactor Theory, Van Nostrand Reinhold Co. New York, 619p., **1970**.
- [11] Driscoll, M. J., Downar, T. J., Pilat, E. E., The Linear Reactivity Model for Nuclear Fuel Management, Amer Nuclear Society, 233p., **1991**.
- [12] Çelikten, O.Ş., *Altigen Geometri Nükleer Reaktör Kor Modeli için Nötronik Hesaplamalar*, Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, **2011**.
- [13] Holland, J.H., Adaptation in natural and artificial systems. MIT Press, MA. **1992**.
- [14] Genetik Algoritmalar, Prof. Dr. Cemalettin KUBAT, <http://www.ebs.sakarya.edu.tr/?upage=ens&page=drs&f=50&b=06&ch=1&yil=2013&disaridan=1&dpage=all&lnKod=39216>, (Ağustos, 2013).
- [15] Chapot, J.L.C., Silva, F.C.D., Schirru, R., A new approach to the use of genetic algorithms to solve the pressurized water reactor's fuel management optimization problem. *Annals of Nuclear Energy* 26, 641–655, **1999**.
- [16] Parks, G.T., Multiobjective pressurized water reactor reload core design by nondominated genetic algorithm search. *Nuclear Science and Engineering* 124, 178–187, **1996**.
- [17] Erdoğan, A., Geçkinli, M., A PWR reload optimisation code (XCore) using artificial neural networks and genetic algorithms, *Annals of Nuclear Energy* 30, 35–53, **2003**.
- [18] Yılmazbayhan, A., *Nuclear Fuel Management Optimization Using Genetic Algorithm*, Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, **1999**.

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Erhan ŞENLİK
Doğum Yeri: Doğanşehir
Medeni Hali: Evli
E-Posta: erhansnk@gmail.com
Adresi: Hacettepe Üniversitesi Kimya Müh. Böl.

Eğitim

Lise: Malatya Lisesi
Lisans: Hacettepe Üniversitesi Nükleer Enerji Mühendisliği

Yabancı Dil ve Düzeyi:

İngilizce / İyi

İş Deneyimi:

(2007)- Kimya Müh. Böl-Uzman

Deneyim Alanları:

Programlama,Ağ-programlam

Tezden Üretilmiş Projeler ve Bütçesi :

-

Tezden Üretilmiş Yayınlar :

-

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar

-