



**T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



[DOKTORA TEZİ]

**[GPU PROGRAMLAMA ile BÜYÜK KÜTLELİ YILDIZ OLUŞUM
BÖLGELERİNİN SPEKTRAL ANALİZİ |**

[Yasemin POYRAZ KOÇAK]

**[DANIŞMAN
Doç. Dr. Selçuk SEVGİN]**

**[II. DANIŞMAN
Prof. Dr. Peter SCHILKE]**

[Bilgisayar Mühendisliği Anabilim Dalı |

[Bilgisayar Mühendisliği Programı |

[İSTANBUL-2021]

Bu çalışma, 18.03.2021 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Programında Doktora tezi olarak kabul edilmiştir.

Tez Jürisi

Doç. Dr. Selçuk SEVGİN (Danışman)
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Prof. Dr. Ahmet SERTBAŞ
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Prof. Dr. Sabri ARIK
İstanbul Üniversitesi-Cerrahpaşa
Elektrik Elektronik Fakültesi

Prof. Dr. Serdar ÖZÖĞÜZ
İstanbul Teknik Üniversitesi
Elektrik-Elektronik Fakültesi

Prof. Dr. Tansel AK
İstanbul Üniversitesi
Fen Fakültesi

20.04.2016 tarihli Resmî Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi-Cerrahpaşa’nın abonesi olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

Bu tez, Spektral Analysis Tools adı altında German ALMA ARC node’a bağlı olan Köln Üniversitesi tarafından ve 2214-A Yurt Dışı Doktora Sırası Araştırma Burs Programı kapsamında 2016/2 döneminde 1059B141600846 başvuru numarası ile TÜBİTAK tarafından desteklenmiştir.

ÖNSÖZ

Öncelikle Yüksek Lisans ve Doktora tezimi hazırladığım ve yurt dışında bulunduğum süre boyunca beni her zaman destekleyen, pes ettim dediğim aşamada ayağa kalkmamı sağlayıp motive eden, hem bilimsel hem de bilimsel olmayan problemlerimin çözümünde danışmanlık yapan ve uzun yıllar boyunca da kendisiyle çalışma temennisinde olduğum çok değerli danışmanım Doç. Dr. Selçuk Sevgen'e en içten dileklerimle teşekkür ederim.

Almanya'da bulunduğum sırada tezimde kullandığım verileri elde etmeme yardımcı olan ve Astrofizik alanındaki deneyimlerini benimle paylaşan, gördüğüm en kibar kişilerden biri olan ikinci danışmanım Prof. Dr. Peter Schilke'ye,

Her daim desteklerini esirgemeyen Prof. Dr. Sabri Arık, Prof. Dr. Tansel Ak, Prof. Dr. Serdar Özoğuz, Doç. Dr. Rüya Şamlı ve Dr. Thomas Möller'e,

Tüm hayatım boyunca her daim yanımda olup, eğitim hayatım boyunca maddi ve manevi desteklerini esirgemeyen aileme, hayattaki en büyük şansım olan eşime ve kızıma,

Son olarak, “GPU Programlama ile Büyük Kütleli Yıldız Oluşum Bölgelerinin Spektral Analizi” başlıklı doktora tez çalışmamı destekleyen TÜBİTAK-BİDEB'e, Köln Üniversitesi'ne, ALMA Projesi çalışanlarına en güzel sevgilerimi sunar en içten dileklerimle teşekkür ederim.

Bu süreç içerisinde hayatımın en güzel ve en birikim dolu yıllarını geçirdim. Yeni insanlar, yeni yerler, yeni araştırma alanları ve önceden hiç farkında olmadığım bilgiler keşfettim. Bu süreçte bazen pes ettim, bazen de ayağa kalktım. Her toparlanışında daha güçlendim, öğrendim, örnek aldım ve örnek olmaya çalıştım. Yıldızlara bakmanın ufkumu ne kadar açtığını fark ettim. Deneyimler bir okuldan daha fazlasını öğretti ve öğrendikçe ben oldum.

Mart 2021

[Yasemin POYRAZ KOÇAK]

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	iv
İÇİNDEKİLER.....	v
ŞEKİL LİSTESİ	viii
TABLO LİSTESİ.....	xi
SİMGE VE KISALTMA LİSTESİ	xii
ÖZET	xiv
SUMMARY	xv
1. GİRİŞ	1
2. GENEL KISIMLAR.....	4
2.1. SPEKTRAL ANALİZ	4
2.1.1. Işık	4
2.1.2. Spektrum.....	5
2.2. ATACAMA BÜYÜK MİLİMETRE/MİLİMETREALTI DİZİSİ (ALMA)	7
2.2.1. İnterferometre Tekniği.....	9
2.3. ASTRONOMİK VERİLERİ MODELLEYEN YAZILIMLAR	10
2.3.1. Görüntü Küçültme ve Analiz Aracı (IRAF)	11
2.3.2. Astronomik Görüntü İşleme Sistemi (AIPS).....	11
2.3.3. Sürekli ve Çizgi Analizi Tek Çanaklı Yazılım (CLASS).....	12
2.3.4. Grenoble Görüntü ve Çizgi Veri Analizi Yazılımı (GILDAS)	12
2.3.5. Geliştirilmiş CASA Çizgi Analizi Yazılım Paketi (XCLASS)	13
2.3.5.1. Eğri Uydurma İşlemi.....	14
2.4. PARALEL PROGRAMLAMA MODELLERİ.....	14
2.4.1. POSIX Thread	15
2.4.2. OpenMP.....	16
2.4.3. Mesaj İletim Arayüzü (MPI)	17
2.4.4. OpenCL	17
2.4.5. CUDA.....	18
2.5. OTOMATİK ÇİZGİ TANIMA YAZILIMI	20
3. MALZEME VE YÖNTEM.....	26
3.1. XCLASS	26

3.1.1.	MOLFIT Dosyası	32
3.1.2.	Iso Ratio Dosyası.....	33
3.1.3.	XCLASS'ın Hızlandırılması İşlemi	34
3.1.3.1.	<i>ModelCalcSpectrum Fonksiyonu</i>	37
3.1.3.2.	<i>InterpolateQGPU Fonksiyonu</i>	39
3.1.3.3.	<i>DetectionEquation Fonksiyonu</i>	41
3.1.4.	Optimizasyon İşlemi.....	42
3.2.	SPEKTRAL ÇİZGİLERİN TESPİTİ.....	45
3.2.1.	Otomatik Çizgi Tanıma Yazılımı	45
3.2.1.1.	<i>Tekli Molekül Uydurma İşlemi</i>	46
3.2.1.2.	<i>Parametre Tahmin Algoritması</i>	46
3.2.1.3.	<i>Tespit Edilen Moleküllerin Kontrolü</i>	50
3.2.1.4.	<i>Tüm Molekül Uydurma İşlemi</i>	51
3.2.2.	Otonom Gauss Ayırıştırma Yazılımı	51
3.2.2.1.	<i>Türevlenebilir Spektroskopi</i>	52
3.2.2.2.	<i>Makine Öğrenmesi Tekniği ile logα Seçimi</i>	54
4.	BULGULAR	56
4.1.	XCLASS'IN HIZLANDIRILMASI İŞLEMİNİN DENEYSEL SONUÇLARI	56
4.2.	SPEKTRAL ÇİZGİLERİN TESPİTİ İŞLEMİNİN SONUÇLARI	62
4.2.1.	Otomatik Çizgi Tanıma Yazılım'ının Test İşlemleri ve Sonuçları.....	62
4.2.2.	Otomatik Çizgi Tanıma Yazılımı Giriş Parametre Değerleri.....	62
4.2.3.	Test İşleminde Kullanılan Spektral Veri ve Veri Tabanı	64
4.2.3.1.	<i>“C₂H₅OH” molekülü için Sonuçlar</i>	65
4.2.3.2.	<i>“CH₃CN” Molekülü için Sonuçlar</i>	68
4.2.3.3.	<i>“CH₃OH” Molekülü için Sonuçlar</i>	70
4.2.3.4.	<i>“HCCCN” Molekülü için Sonuçlar</i>	71
4.2.3.5.	<i>“SO₂” Molekülü için Sonuçlar</i>	72
4.2.3.	AGD Yazılımının Sonuçları	74
4.2.3.1.	<i>“C₂H₅OH” Molekülü için Sonuçlar</i>	75
4.2.3.2.	<i>“CH₃CN” Molekülü için Sonuçlar</i>	77
4.2.3.3.	<i>“CH₃OH” Molekülü için Sonuçlar</i>	78
4.2.3.4.	<i>“HCCCN” Molekülü için Sonuçlar</i>	80
4.2.3.5.	<i>“SO₂” Molekülü için Sonuçlar</i>	80

4.2.3. Otomatik Çizgi Tanıma Yazılımı ve AGD Algoritmalarının Karşılaştırılması.....	81
5. TARTIŞMA VE SONUÇ	83
KAYNAKLAR.....	88
ÖZGEÇMİŞ.....	91
a. 2017 – 2019, Spectral analysis tools, Poyraz Koçak Y., Sevgen S., ALMA Projesi.	Hata! Yer işareti tanımlanmamış.
b. 2017 – 2018, Detecting and counting people using realtime directional algorithms implemented by compute unified device architecture, (Yürütücü) Sevgen S., Poyraz Koçak Y., İstanbul Üniversitesi-Cerrahpaşa BAP Projesi.	Hata! Yer işareti tanımlanmamış.



ŞEKİL LİSTESİ

	Sayfa No
Şekil 2.1: Işığın dalga boyu zaman grafiği. (Karaahmet, 2017)	4
Şekil 2.2: Elektromanyetik dalga çeşitleri. (Anıl, 2016).....	5
Şekil 2.3: Işığın soğurma ve ışıma spektrumu. (Uslu, 2019).....	6
Şekil 2.4: Bazı atomların ışıma spektrumu. (Uslu, 2019).....	7
Şekil 2.5: ALMA projesinde kullanılan yüksek hassasiyete sahip radyo teleskopların görüntüsü. (Mühle ve diğ., 2013)	8
Şekil 2.6: ALMA cihazlarının konumunu gösteren harita. (Mühle ve diğ., 2013)	9
Şekil 2.7: XCLASS yazılımının yapısı. (Sánchez-Monge, 2018).....	13
Şekil 2.8: NGC6334-I adlı bulutsunun 579-582 GHz frekans aralığındaki spektrumuna uygulanan eğri uydurma işlemi. (Sánchez-Monge, 2018).....	14
Şekil 2.9: CUDA mimarisi ve bellek modeli. (Diaz ve diğ., 2012)	20
Şekil 2.10: Orion KL adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektrumdaki moleküler bantlar. (Tennyson, 2005).....	22
Şekil 2.11: Frekans bandına ve Radyo teleskobun çapına bağlı olarak Gauss eğrisinin değişimi. (RF Wireless World, 2017).....	25
Şekil 3.1: Teleskoba gelen bir ışının içindeki çekirdek katmanlarının dağılımı. (Möller ve Schilke, 2017)	26
Şekil 3.2: Siyah halka ile belirtilen Gauss şeklindeki bir ışının içindeki çekirdek ve ön plan katmanlarının dağılımı. (Möller ve Schilke, 2017)	31
Şekil 3.3: Örnek MOLFIT dosyası. (Möller ve Schilke, 2017)	33
Şekil 3.4: Toz geçirgenlik ve toz sıcaklık değerleri eklenmiş örnek MOLFIT dosyası. (Möller ve Schilke, 2017).....	33
Şekil 3.5: Module_myXCLASS_Core.f90 dosyasında bulunan fonksiyonlar.	35
Şekil 3.6: Module_myXCLASS_GPU.f90 dosyasında bulunan fonksiyonlar.	36
Şekil 3.7: ModelCalSpectrum giriş değerleri.....	38
Şekil 3.8: InterpolateQGPU fonksiyonunun cihaz dizileri.	40

Şekil 3.9: DetectionEquation fonksiyonunun cihaz dizileri.....	42
Şekil 3.10: Sayfalanabilir bellekten sunucudan cihaza veri transferi. (Ruetsch ve Fatica, 2014).....	43
Şekil 3.11: Sayfa kilitli bellekte sunucudan cihaza veri transferi. (Ruetsch ve Fatica, 2014).....	44
Şekil 3.12: Otomatik Çizgi Tanıma yazılımı akış şeması.	45
Şekil 3.13: Parametre Tahmin Algoritması akış şeması.	47
Şekil 3.14: CH ₃ OH adlı molekülün v=0 km/s'de geçiş frekans spektrum penceresi. (Möller, 2018).....	48
Şekil 3.15: Şekil 3.14'te gösterilen CH ₃ OH'nin geçiş frekans spektrum penceresinin merkezi hızlarının histogramı. (Möller, 2018)	49
Şekil 3.16: Türevlenebilir Spektrum örneği.....	53
Şekil 4.1: CPU platformu üzerinde ArH + molekülünün eğri uydurma işlemi (solda) ve chi2 hata vektörü (sağda).....	58
Şekil 4.2: GPU platformu üzerinde ArH + molekülünün eğri uydurma işlemi (solda) ve chi2 hata vektörü (sağda).....	58
Şekil 4.3: 10 döngü ile yapılan modelleme işlemi.	59
Şekil 4.4: 50 döngü ile yapılan modelleme işlemi.	60
Şekil 4.5: 100 döngü ile yapılan modelleme işlemi.	60
Şekil 4.6: GPU ve CPU platformu üzerinde döngü sayısının toplam çalışma süresine etkisinin grafiksel gösterimi.	61
Şekil 4.7: Otomatik Çizgi Tanıma Yazılımı ile "C ₂ H ₅ OH" molekülüne ait tespit edilen spektral çizgilerin grafiksel gösterimi.	66
Şekil 4.8: Otomatik Çizgi Tanıma Yazılımı ile "CH ₃ CN" molekülüne ait tespit edilen spektral çizgilerin grafiksel gösterimi.	69
Şekil 4.9: Otomatik Çizgi Tanıma yazılımı ile "CH ₃ OH" molekülüne ait tespit edilen spektral çizgilerin grafiksel gösterimi.	70
Şekil 4.10: Otomatik Çizgi Tanıma Yazılımı ile "HCCCN" molekülüne ait tespit edilen spektral çizginin grafiksel gösterimi.	72
Şekil 4.11: Otomatik Çizgi Tanıma Yazılımı ile "SO ₂ " molekülüne ait tespit edilen spektral çizgilerin grafiksel gösterimi.	73
Şekil 4.12: AGD yazılımı ile SgrB2 adlı bölgede tespit edilen tüm spektral çizgiler.	74

Şekil 4.13: AGD yazılımı ile “C ₂ H ₅ OH” adlı moleküle ait spektral çizgiler.	76
Şekil 4.14: AGD yazılımı ile “CH ₃ CN” adlı moleküle ait spektral çizgiler.	77
Şekil 4.15: AGD yazılımı ile “CH ₃ OH” adlı moleküle ait spektral çizgiler.	79
Şekil 4.16: AGD yazılımı ile “HCCCN” adlı moleküle ait tespit işleminin grafiksel gösterimi.	80
Şekil 4.17: AGD yazılımı ile “SO ₂ ” adlı moleküle ait spektral çizgiler.	81



TABLO LİSTESİ

Sayfa No

Tablo 3.1: Tesla K20 veri transfer hızı. (Ruetsch ve Fatica, 2014).....	44
Tablo 4.1: Tesla K20m GPU ekran kartının özellikleri.....	57
Tablo 4.2: Deneysel sonuçlar ve performans kazancı.	57
Tablo 4.3: Diğer test fonksiyonlarının CPU ve GPU platformu üzerinde çalışma süreleri ve performans kazancı.	57
Tablo 4.4: GPU ve CPU platformu üzerinde döngü sayısının toplam çalışma süresine etkisi ve döngü sayısına bağlı olarak toplam çalışma süresinin artış oranı.	61
Tablo 4.5: Otomatik Çizgi Tanıma Yazılımı ile " C_2H_5OH " molekülüne ait tespit edilen spektral çizgilerin frekans aralığı, konum ve genlik değerleri.	66
Tablo 4.6: Otomatik Çizgi Tanıma Yazılımı ile " CH_3CN " molekülüne ait tespit edilen spektral çizgilerin frekans aralığı, konum ve genlik değerleri.	69
Tablo 4.7: Otomatik Çizgi Tanıma Yazılımı ile " CH_3OH " molekülüne ait tespit edilen spektral çizgilerin frekans aralığı, konum ve genlik değerleri.	71
Tablo 4.8: Otomatik Çizgi Tanıma Yazılımı ile " $HCCCN$ " molekülüne ait tespit edilen spektral çizginin frekans aralığı, konum ve genlik değeri.....	72
Tablo 4.9: Otomatik Çizgi Tanıma Yazılımı ile " SO_2 " molekülüne ait tespit edilen spektral çizgilerin frekans aralığı, konum ve genlik değeri.	73
Tablo 4.10: AGD yazılımı ile SgrB2 adlı bölgede tespit edilen tüm spektral çizgilerin ait FWHM, frekans ve genlik değerleri.	74
Tablo 4.11: AGD yazılımı ile " C_2H_5OH " adlı moleküle ait tespit edilen her bir spektral çizgiye ait FWHM, frekans ve genlik değerleri.	76
Tablo 4.12: AGD yazılımı ile " CH_3CN " adlı moleküle ait tespit edilen her bir spektral çizgiye ait FWHM, frekans ve genlik değerleri.	78
Tablo 4.13: AGD yazılımı ile " CH_3OH " adlı moleküle ait tespit edilen her bir spektral çizgiye ait FWHM, frekans ve genlik değerleri.	79
Tablo 4.14: AGD yazılımı ile " SO_2 " adlı moleküle ait tespit edilen her bir spektral çizgiye ait FWHM, frekans ve genlik değerleri.	81
Tablo 4.15: Otomatik Çizgi Tanıma ve AGD algoritmaları ile her bir moleküle ait ve toplam tespit edilen spektral çizgilerin sayısı.....	82

SİMGE VE KISALTMA LİSTESİ

Kısaltmalar	Açıklama
AGD	: Autonomous Gaussian Decomposition
AIPS	: Astronomical Image Processing System
ALMA	: Atacama Large Millimeter/Submillimeter Array
API	: Application Programming Interface
APU	: Accelerated Processing Units
ARC	: ALMA Regional Center
CARMA	: Combined Array for Research in Millimeter-wave Astronomy
CASA	: Common Astronomical Software Application
CCD	: Charge Coupled Device
CDMS	: Cologne Database for Molecular Spectroscopy
CHARA	: Center for High Angular Resolution Astronomy
CLASS	: Continuum and Line Analysis Single-dish Software
CPU	: Central Processing Unit
CUDA	: Compute Unified Device Architecture
DMA	: Direct Memory Access
FWHM	: Full Width Half Maximum
GILDAS	: Grenoble Image and Line Data Analysis Software
GNU	: General Public License
GPGPU	: General-Purpose Computing on Graphics Processing Units
GPU	: Graphics Processing Unit
HPP	: Heterogeneous Parallel Programming
ILP	: Instruction-Level Parallelism
IRAF	: Image Reduction and Analysis Facility
IRAM	: Institute For Radio Astronomy Millimeter
JPL	: Jet Propulsion Laboratory
LTE	: Local Thermodynamic Equilibrium
MIC	: Many Integrated Core
MPI	: Message Passing Interface
MROI	: Magdalena Ridge Observatory Interferometer
NOAO	: National Optical Astronomy Observatory

NOEMA	: NOrthern Extended Millimeter Array
NPOI	: Navy Prototype Optical Interferometer
OpenCL	: Open Computing Language
OpenMP	: Open Multi-Processing
OSF	: Operations Support Facility
POSIX	: Portable Operating System Interface
RAM	: Random Access Memory
SgrB2	: Sagittarius B2
SMA	: Submillimeter Array
XCLASS	: Extended CASA Line Analys Software Suit
VLA	: Very Large Array
VLT	: Very Large Telescope

ÖZET

DOKTORA TEZİ

GPU PROGRAMLAMA İLE BÜYÜK KÜTLELİ YILDIZ OLUŞUM BÖLGELERİNİN SPEKTRAL ANALİZİ

Yasemin POYRAZ KOÇAK

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Selçuk SEVGİN

II. Danışman: Prof. Dr. Peter SCHILKE

Astronomi, çok büyük boyutlu veriler ile çalışmak zorunda olan bir bilim dalıdır. Bu verilerin işlenmesi ve bilgi çıkarımı çok uzun zaman aldığı için bu verileri modelleyen bilgisayar yazılımlarının performansı çok önemlidir. Tez kapsamında gerçekleştirilen ilk çalışmada, ALMA'dan elde edilen verileri modellemek amacı ile Köln Üniversitesi tarafından geliştirilen XCLASS'ın hesap yoğun kısımları, GPU programlama tekniği ile hızlandırılmıştır. Galaksideki en büyük ve karmaşık büyük kütleli yıldız oluşum bölgelerinden biri olan, Sagittarius B2 adlı bölgeden elde edilen spektral veriler ile, XCLASS'ın CPU ve GPU versiyonları test edilmiş, performansları karşılaştırılmıştır. Elde edilen sonuçlar, GPU programlama tekniğinin, büyük boyutlu astronomik verilerin modellenmesinde kullanılan yazılımlarda yüksek performans kazancı sağladığını göstermiştir. Gökyüzünden alınan verilerden elde edilen spektral verilerde bulunan her bir çizginin varlığının ve yükseklik, frekans, genlik gibi parametrelerinin tespiti, hangi molekül, atom veya bileşenlerden oluştuğunun belirlenmesi, bölgenin yapısının çıkarılması çok önemlidir. Tez kapsamında gerçekleştirilen ikinci çalışmada, XCLASS'a entegre edilmek üzere SgrB2 adlı bölgeden elde edilen spektral verilerdeki tespit edilmesi zor olan, iç içe geçmiş çizgilerin tespiti işlemi için Otomatik çizgi tanıma yazılımı geliştirilmiştir. Gauss paketi içerisinde AGD yazılımı ile aynı verilere ait spektral çizgilerin tespiti işlemi gerçekleştirilmiş ve başarı oranları karşılaştırılmıştır. Elde edilen sonuçlar, Otomatik Çizgi Tanıma yazılımının spektral çizgilerin tespiti işleminde daha iyi sonuçlar sunduğu görülmüştür. Bu kadar büyük ve karmaşık bir alandan elde edilen verilerin başarılı bir şekilde analiz edilmesi, evrenin farklı bölgelerinde de analiz yapılabileceği anlamına gelecektir, böylece karanlık evrenin fiziksel ve kimyasal yapısı hakkında daha fazla bilgi edinmek mümkün olacaktır.

Mart 2021, 105 sayfa.

Anahtar kelimeler: GPU Programlama Tekniği, Spektral Analiz, Astroinformatik, XCLASS

SUMMARY

Ph.D. THESIS

**SPECTRAL ANALYSIS OF MASSIVE STAR FORMING REGION BY GPU
PROGRAMMING**

Yasemin POYRAZ KOÇAK

Istanbul University-Cerrahpasa

Institute of Graduate Studies

Department of Computer Engineering

Supervisor : Assoc. Prof. Dr. Selçuk SEVGİN

Co-Supervisor : Prof. Dr. Peter SCHILKE

Astronomy is a science that has to work with very large data. Since processing of these data and extracting useful information from these data take a long time, the performance of software that models these data is very important. In the first study conducted within the scope of this thesis, compute-intensive parts of XCLASS, developed by the University of Cologne to model the data obtained from ALMA, have been accelerated by using GPU programming technique. CPU-GPU versions of XCLASS have been tested on the spectral data obtained from star formation region SgrB2 which has a mass of about 3 million of the solar mass and is one of the largest and complex high mass star formation regions and performance comparison have been made. The results show that GPU programming technique provides significant performance gain in software used for modeling large scale astronomical data. Determination of existence and parameters such as height, frequency and amplitude of each line in the spectral data obtained from the sky has great importance for determination of the structure of the region. In the second study conducted within the scope of this thesis, Automatic Line Identification software has been developed for detection of nested lines in the spectral data obtained from SgrB2 to be integrated into XCLASS. Spectral lines of the same data have been determined by using AGD algorithm in Gaussian package and the success rates have been compared with Automatic Line Identification software. The results show that the Automatic Line Identification software is successful on detecting of spectral lines. Successful analysis of large and complex area means that analysis can be made in different parts of the universe, thus, it will be possible to learn more about the physical and chemical structure of the dark universe.

March 2021, 105 pages.

Keywords: GPU programming technique, Spectral analysis, Astroinformatics, XCLASS

1. GİRİŞ

Astronomi, gök cisimlerini (yıldızlar, gezegenler, kuyruklu yıldızlar ve galaksiler gibi) ve dünya atmosferi dışında (kozmetik fon ışıması gibi) ortaya çıkan olayları inceleyen en eski bilim dallarından biridir (Unsöld ve Baschek, 1991). Dünyada hızla artan nüfus ile gelecekte kaynakların tükenmesi tehlikesi ile karşı karşıya kalan insanlık başka gezegenlerde yaşamın var olup olmadığını araştırmaktadır. Yıldızlar evrende bildiğimiz her şeyi ortaya çıkaran cisimlerdir. Yıldızları anlamadan gezegenlerin hatta insanların nereden geldiğini anlamak oldukça zordur. Başka gezegenlerdeki yaşamın var olup olmadığını anlamak için önce yıldızları anlamak gerekir. Yıldızları incelemek için ise NOEMA (Winters ve diğ., 2019), Plateau de Bure (Guilloteau ve diğ., 1999), CARMA (Mundy, 2013), SMA (Gurwell, 2020), VLA (NRAO, 2009) ve ALMA (Schieven, 2015) gibi ileri tekniklere sahip gelişmiş gözlem araçlarına ihtiyaç vardır. Bu tür gelişmiş gözlem araçlarından bir gecede 30TB'a varan çok büyük boyutlarda veri elde edilmektedir. Bundan sonraki 10 yılda gözlem araçlarının gelişmesi ile birlikte yapılan çalışmalarda elde edilen verilerin boyutunun yüzlerce petabyte'a ulaşması beklenmektedir. Bilgi işlem gücü, depolama ve bant genişliği alanlarındaki istikrarlı artışlar ve gittikçe artan miktarda verinin verimli bir şekilde nasıl ele alınacağı ve işleneceği sorusu ile astronomi alanı baş başa kalmıştır. Bu nedenle gelecekte gelişmiş gözlem araçları ile elde edilen yüzlerce petabytelar boyutundaki verilerin sadece işlenebilir olanlarını almak gerekmektedir. Bu durum astronomi alanının bilgisayarların üstün hesaplama gücüne ve bilgisayar biliminin yöntemlerine olan ihtiyacını artırmıştır. Geliştirilen tüm yöntemler yeni bir çalışma konusu olan "Astroinformatik" alanını ortaya çıkarmıştır.

Astroinformatik, astronomi ve veri bilimi, makine öğrenimi, bilişim, iletişim teknolojilerinin kombinasyonundan oluşan disiplinler arası bir çalışma alanıdır. Astroinformatik, veri odaklı astronomi alanında veri bilimi, makine öğrenimi, istatistik araçları, yöntemleri ve uygulamalar geliştirmeyi amaçlamıştır.

Astronomi alanında bu denli büyük boyutlu verilerin işlenmesi ve bu verilerden bilgi keşfi ve öğrenme çok uzun zaman aldığı için bu verileri modelleyen yazılımların performansı çok önemlidir. Bu nedenle astroinformatik, gökbilimciler için yeni yöntemler ve yazılımlar

geliştirmeye yardımcı olmanın yanı sıra astronomi alanında hızla artan veri miktarının işlenmesini ve analizini kolaylaştıran yazılımların performansını artırmayı da amaçlamıştır.

Çok büyük boyutlu astronomik verileri işlemek amacı ile kullanılan yazılımlarda CPU'lar tek başına gerekli performansı sağlamakta yetersiz kalmaktadır. Bu sorun, OpenMP (Jang, 2008), MPI (Arimilli ve Siegel, 2002), SMP (Diaz, 2012) ve CUDA (Sandersand, Kandrot, 2010) gibi paralel programlama için kullanılan paralel işlem yapabilme özelliğine sahip yöntemlerin ve NVIDIA ekran kartı gibi çok çekirdekli cihazların kullanılmasıyla çözülebilir hale gelmiştir. OpenMP ve SMP çok çekirdekli işlemci üzerinde kurulu paralel programlama teknikleridir. MPI ise kullanılabilmesi için öncelikle sisteme yüklenmesi gereken, CPU içerisindeki çekirdekler arası mesajlaşmayı sağlayan bir kütüphanedir.

CUDA, NVIDIA tarafından geliştirilmiş, bir paralel hesaplama platformu ve programlama modelidir. CUDA, geliştirilen uygulamaların yürütülmesi için CPU'ya göre çok daha fazla sayıda çekirdeğe sahip olan GPU'nun sanal komut setine ve paralel hesaplama öğelerine doğrudan erişim sağlar. Bu sayede CUDA, GPU'ların gücünden yararlanarak yoğun işlem gerektiren uygulamaların hızlandırılmasını daha mümkün kılmaktadır.

Tez çalışmasının ilk kısmında, Köln Üniversitesi'nden Prof. Dr. Peter Schilke ve Dr. Thomas Möller tarafından geliştirilmiş, ALMA cihazlarından alınan astronomik verileri modelleme amacı ile kullanılan XCLASS adlı yazılımının hesap yoğun kısımlarının GPU programlama tekniği kullanılarak hızlandırılması işlemi gerçekleştirilmiştir. Hızlandırma işleminin yapıldığı bölüm Fortran programlama dili kullanılarak geliştirildiği için hızlandırma işlemi için CUDA Fortran programlama dili kullanılmıştır ve elde edilen sonuçların, gelecekte daha da büyük boyutlu verilerle çalışmak zorunda olan ALMA projelerine iyi bir kaynak olacağı görülmüştür.

Mutlak sıfırdan daha sıcak olan tüm cisimler elektromanyetik ışıyım yayar ve evrendeki nesneleri incelemek için yaydıkları ışık dışında başka veri kaynağı yoktur. Optik bölgede prizma veya difraksiyon şebekesi (diffraction grating) gibi optik bileşenlerin teleskoplarla birlikte kullanılması sayesinde spektrum elde edilir. Her atomun ve molekülün kendine özgü spektrumu olduğundan, bir gökyüzü bölgesinin spektrumunun özelliklerinin ortaya konması o bölgenin radyal hızı, sıcaklığı, molekül miktarı gibi birçok fiziksel ve kimyasal bilgi elde edilmesinde önemli rol oynar (Tennyson, 2005). Bu tür özelliklerin belirlenmesi, spektrumdaki tespit edilmesi zor, iç içe geçmiş her çizginin varlığının ve parametrelerinin belirlenmesi ile

gerçekleşir. Bu nedenle, spektrum analizi, gökyüzündeki herhangi bir cismin fiziksel özellikleri hakkında bilgi edinebilmek için kullanılan en önemli yöntemlerden biridir.

Tez çalışmasının ikinci aşamasında da XCLASS adlı yazılıma entegre edilmek üzere ALMA cihazları ile SgrB2 (Schilke, 2011) adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral verilerdeki iç içe geçmiş, tespit edilmesi zor her bir çizgiyi ve bu çizginin hangi frekans aralığında bulunduğu ve genlik değerini tespit eden Otomatik Çizgi Tanıma yazılımı geliştirilmiştir. Otomatik Çizgi Tanıma yazılımının başarı oranı Gauss paketi ile karşılaştırılmıştır. Elde edilen sonuçlarda daha fazla sayıda çizgi tespit edildiği görülmüştür.

Tez çalışmasının ilerleyen bölümlerinin içerikleri şu şekildedir: “Genel Kısımlar” bölümünde, ALMA, XCLASS, GPU Programlama Tekniği, Otomatik Çizgi Tanıma yazılımı hakkında temel bilgiler verilmiştir. “Malzeme ve Yöntemler” bölümünde, XCLASS’ın hesap yoğun kısımlarının GPU programlama tekniği kullanılarak nasıl hızlandırıldığı, XCLASS’a entegre edilmek amacı ile spektral verilerdeki her bir çizginin tespiti işlemini gerçekleştiren Otomatik Çizgi Tanıma Yazılımı için nasıl bir algoritma geliştirildiği ve son olarak Gauss paketinin spektral verilerdeki her bir çizginin tespiti işlemi için kullandığı yöntem hakkında bilgi verilmiştir. “Bulgular” bölümünün birinci kısmında XCLASS’ın hesap yoğun bölümlerinin GPU programlama tekniği ile hızlandırılması ile elde edilen sonuçlardan ve sonuçların iyileştirilmesi için gerçekleştirilen optimizasyon işleminden bahsedilmiştir. “Bulgular” bölümünün ikinci kısmında ise, SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden alınan spektral veriler kullanılarak, XCLASS’a entegre edilmek üzere geliştirilen Otomatik Çizgi Tanıma Yazılımı ve Gauss paketi test edilmiştir. Bulunan sonuçlar ile geliştirilen Otomatik Çizgi Tanıma Yazılımının etkinliğinin karşılaştırmaları yapılmıştır. Son olarak “Tartışma ve Sonuç” bölümünde, elde edilen sonuçların değerlendirilmesi ve tez sonrası hedeflenen çalışmalardan bahsedilmektedir. |

2. GENEL KISIMLAR

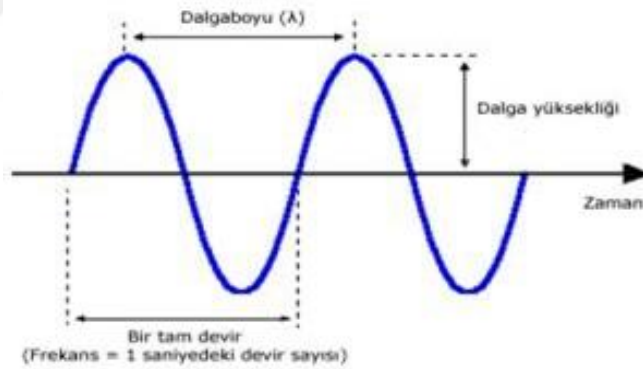
Bu bölümde spektral analiz, ALMA, astronomik verileri modelleyen yazılımlar, paralel programlama modelleri ve otomatik çizgi tanıma yazılımı hakkında bilgiler verilecektir.

2.1.SPEKTRAL ANALİZ

2.1.1. Işık

Evrendeki nesneler hakkında bilgi sahibi olmak için elimizdeki tek kaynak yaydıkları ışıktır. Astronomların gök cisimleri hakkındaki soruları yanıtlamak amacıyla ışığı nasıl analiz ettiklerini anlamak için, ışığın kendisi hakkında biraz bilgi sahibi olmak gerekir.

Işık, bir dalga gibi davranarak enerji taşır. Şekil 2.1’de gösterildiği gibi dalganın iki ardıl tepesi arasındaki mesafe dalga boyu (λ) olarak adlandırılır. Dalga sürekli hareket halinde olduğundan bir saniyedeki dalganın devir sayısı ise frekansı verir.

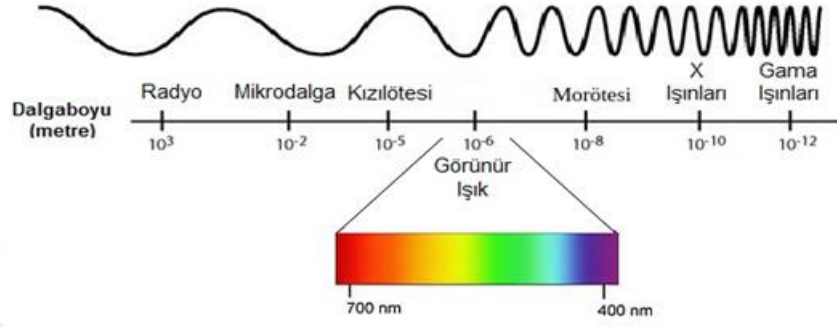


Şekil 2.1: Işığın dalga boyu zaman grafiği. (Karaahmet, 2017)

Beyaz ışık aynı hızda farklı dalga boylarına sahip ışıklardan oluşur. Dalga boyu farklı olan ışıklar farklı renkler olarak algılanır. Kırmızıdan mora doğru dalga boyunda kısalma olur. Örneğin kırmızı ışığın dalga boyu 700 nm iken mor ışığın dalga boyu 400 nm’dir. Işığın hızı (c) yaklaşık 300.000km/sn’dir. Eğer bir ışık ışını Dünya’nın yüzeyinde çepeçevre seyahat edebilseydi, bu küreyi saniyede 7 sefer dolanırdı. (Tennyson, 2005)

2.1.2. Spektrum

Fiziksel bir terim olarak spektrum, bir dalga olarak elektromanyetik ışınının şiddetinin dalga boyuna göre dağılımıdır. Elektromanyetik ışınının dalga boyu Şekil 2.2’de görüldüğü gibi, en kısa dalga boylu gama ışınları ve X ışınlarından başlayarak, ultraviyole, görünür ışık ve kızılötesi ışınlardan sonra en uzun dalga boylu radyo dalgalarına kadar uzanır.

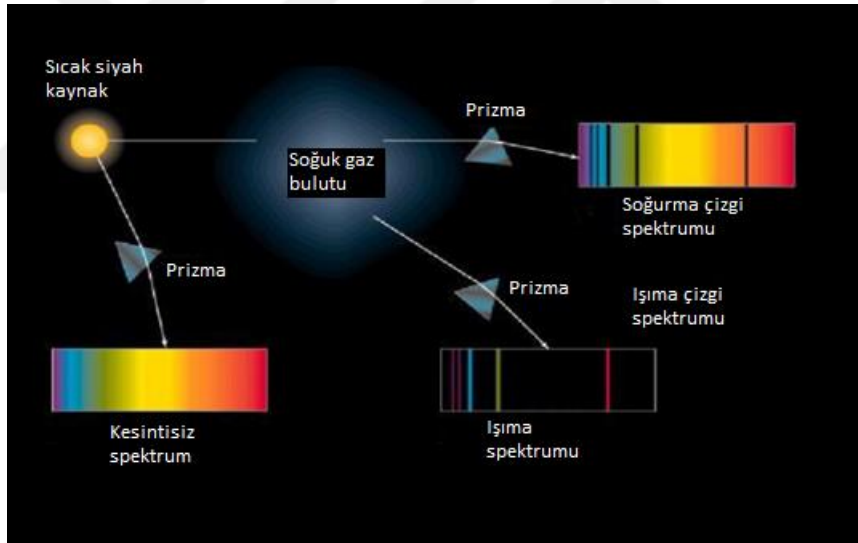


Şekil 2.2: Elektromanyetik dalga çeşitleri. (Anıl, 2016)

Mutlak sıfırdan daha sıcak olan tüm cisimler elektromanyetik ışınım yayar ve evrendeki nesneleri incelemek için yaydıkları ışık (elektromanyetik ışınım) ve gravitasyonel dalgalar dışında başka veri kaynağı yoktur. Elektromanyetik spektrumun mor ötesi, görsel bölge ve kızılötesi dalga boyları optik teleskoplarla gözlenebilir. Ancak gama, X ışın ve radyo teleskopların yapıları optik teleskoplardan farklıdır. Hatta radyo teleskoplar da kendi içlerinde çok farklı yapılarda olabilir. Optik bölgede prizma veya difraksiyon şebekesi (diffraction grating) gibi optik bileşenlerin teleskoplarla birlikte kullanılması sayesinde spektrum elde edilebilir. Bu amaçla kullanılan spektrograflarda gözlenen nesnenin teleskobun odak düzlemindeki görüntüsü spektrograf yarığından ve kolimatörden geçirilerek difraksiyon elemanına (prizma veya şebeke) gönderilir ve bunun oluşturduğu spektrum bir kamera tarafından iki boyutlu bir görüntü olarak kaydedilir. Bu görüntünün dalga boyu eksenine göre oluşturulan şiddet haritası ile spektrum içindeki yapılar dalga boyu-şiddet grafiği halinde elde edilerek incelenebilir. Halbuki çok az görüntüleme elemanlı olanlar hariç, genelde radyo teleskoplar bir cisim bütünü olarak görüntüleyemez. Bunu yerine, gözlemlenen nokta, belli bir frekans aralığında ve belli frekans adımlarında taranarak, her bir frekansta kaynaktan gelen ışınım şiddetini kaydeder. Sonuçta, gözlenen nokta için elde edilen frekans-şiddet grafiği, o noktanın spektrumu olur. Eğer bu tarama komşu çok sayıda gözlem noktası için yapılırsa, bunların her frekans için haritalanması sayesinde, belli bir alanın her frekans için ayrı ayrı görüntüsü elde edilir. Bu görüntüler, astrofizik açısından çok verimli verilerdir. Ek olarak,

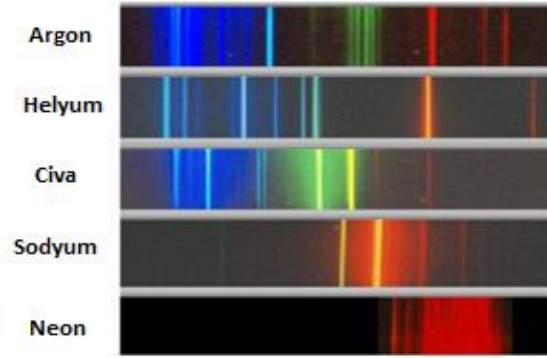
Güneş'in etkisi sebebiyle optik gözlemler sadece gece saatlerinde yapılırken, radyo gözlemleri gündüz saatlerinde de yapılabilir. Yıl boyunca daha çok kaynağın gözlenebilmesini sağladığından, bu da radyo gözlemlerinin önemli bir avantajıdır.

Atom veya molekül yapısı içindeki elektronların enerji seviyelerindeki değiřtirmelerine genel olarak "geçiř" adı verilir ve bu geçiřler atom veya moleküllerin fotonları soğurmaları sonucunda meydana geldiğinden- Kirschhoff yasaları ile tanımlandığı üzere spektrum çizgileri (sürekli spektrum üzerinde soğurma çizgileri veya karanlık zeminde emisyon çizgileri) oluşur. Örneğın Şekil 2.3'te görüldüğü gibi beyaz ışık bir gaza gönderilir ve gazla etkileřime girdikten sonra prizmadan geçerse, renk dağılımı kesintili olur. Bu durum gazın bu eksik renkleri soğurduğunu gösterir ve bu olaya soğurma spektrumu denir. Gazın kendi yaydığı ışık prizmadan geçerse, az önceki şeklin eksiklerini kapatacak şekilde bir spektrum elde edilir. Bu spektruma ışıma spektrumu (emisyon) denir.



Şekil 2.3: Işığın soğurma ve ışıma spektrumu. (Uslu, 2019)

Her atom ve molekülün bulunduğu ortamın fiziksel koşullarına bağılı olarak kendine özgü emisyon ve soğurma çizgileri oluşur (Şekil 2.4). Bu sebeple, gök cisimlerinin spektrumlarındaki söz konusu spektral çizgilerin incelenmesi sayesinde, gök cisimlerinin fiziksel özellikleri anlaşılabilir (Uslu, 2019). Bu nedenle, spektrum analizi, herhangi bir nesnenin fiziksel özellikleri hakkında bilgi edinebilmek için kullanılan en önemli yöntemlerden biridir.



Şekil 2.4: Bazı atomların ışıma spektrumu. (Uslu, 2019)

Belli geçişler ancak belli enerjiye sahip fotonların soğurmasıyla olur. Elektron daha sonra eski seviyesine geri döner ve aynı enerjiye sahip bir foton yayınlar. Bazen birkaç farklı seviyeden adım adım geri döner ve her seferinde belli enerjilere sahip fotonlar yayınlar. Bunlar Gauss eğrilerinin genişlemesine katkıda bulunurlar. Laboratuvarıda üretilen atom ya da molekül çizgileri sakin (rest) haldedir, diğer bir deyişle hızları yoktur. Ama astronomik bir ortamdan kaynaklanıyorsa, bir hızları olur ve bu durum çizgilerde dalga boyu kayması ve çizgi genişlemesi olarak görülür. Bütün bunların hepsi ve daha fazlası birleşince molekül bantları oluşur. Şekil 2.4'te gösterildiği gibi her atomun ve molekülün kendine özgü spektrumu olduğundan, bir gökyüzü bölgesinin spektrumunun özelliklerinin ortaya konması o bölgenin radyal hızı, sıcaklığı, molekül miktarı gibi birçok fiziksel bilgi elde edilmesinde önemli rol oynar (Tennyson, 2005).

2.2. ATACAMA BÜYÜK MİLİMETRE/MİLİMETREALTI DİZİSİ (ALMA)

ALMA, Kuzey Amerika, Kanada, Almanya, İtalya, İsveç, Fransa, Hollanda, İngiltere, Portekiz, Çek Cumhuriyeti, Japonya, Tayvan ve Güney Kore'nin dahil olduğu astronomi alanında en büyük uluslararası bilimsel projelerden biridir. Şili- And Dağları'nın Chajnantor ovasında yaklaşık 5000m yükseklikte ve -23° enlemde yer alan ALMA, inşası henüz tamamlanmamasına rağmen, elli adet 12m çapında (Atacama Compact Array-Morita Dizisi olarak da bilinir), birbirine yakın olarak bulunan on iki adet 7m çapında ve dört adet 12m çapında (Toplam güç dizisi olarak da bilinir) olmak üzere toplamda 66 adet yüksek hassasiyete sahip dünyanın en güçlü radyo teleskoplarına sahiptir (Şekil 2.5).



Şekil 2.5: ALMA projesinde kullanılan yüksek hassasiyete sahip radyo teleskopların görüntüsü.
(Mühle ve diğ., 2013)

ALMA cihazlarının bakım ve onarımların işlemleri, oksijen yetersizliğinden dolayı 2900 m yükseklikte bulunan kontrol odası, ofisler, laboratuvarlar, personel konutları ve yüklenici kampın bulunduğu Operasyon Destek Tesisi (OSF- Operations Support Facility) tarafından yapılmaktadır. Şekil 2.6’da ALMA cihazlarının bulunduğu bölgenin haritası gösterilmektedir.

Gözlemevi ile astronomik topluluk arasındaki ara yüz ALMA Bölgesel Merkezleri (ARC- ALMA Regional Centers) aracılığı ile yapılmaktadır. Kuzey Amerika ve Tayvan ülkelerinin dahil olduğu Kuzey Amerika-ARC (North American ARC), Almanya Merkezli olmak üzere İtalya, İsveç, Fransa, Hollanda, Birleşik Krallık, Portekiz ve Çek Cumhuriyeti devletlerinin dahil olduğu Avrupa-ARC (EU-ARC) ve Tokyo, Japonya, Güney Kore ülkelerinin dahil olduğu Doğu Asya-ARC (EA-ARC)) olmak üzere üç ARC bulunmaktadır. ALMA cihazlarından elde edilen verilere ihtiyacı olan bu ülkeler dışında bulunan araştırmacılar çalışmalara dahil olmak için bu üç merkezden biri ile iletişime geçip çalışmalarını yürütülebilir. Bu tez çalışması ALMA projesinin bir parçasıdır ve ALMA Projesi’ne Avrupa-ARC’ye bağlı olan Köln Üniversitesi aracılığı ile dahil olunmuştur.



Şekil 2.6: ALMA cihazlarının konumunu gösteren harita. (Mühle ve diğ., 2013)

2.2.1. İnterferometre Tekniği

Muazzam boyuttaki evreni gözlemlemek ve daha fazla veri ve faydalı bilgiler elde etmek için kullanılan cihazların çapı her geçen gün artmaktadır. Her ne kadar cihazların çapının artması ile ışık toplama gücü artsa da bu durum, cihazların maliyetinin artmasına ve bakım sorunlarına yol açmaktadır. Bu problemleri çözmek için büyük ölçekli cihazlar kullanmak yerine birden fazla cihazın belirli bir sırayla birleştirilmesiyle oluşturulan astronomik interferometre yöntemi kullanılmaktadır.

İnterferometre, yıldızlar, bulutsular ve galaksiler gibi astronomik nesnelerin interferometri tekniği vasıtasıyla daha yüksek çözünürlüklü görüntülerini elde etmek için tek bir teleskop olarak birlikte çalışan ayrı teleskoplar veya radyo teleskop antenleri dizisidir. İnterferometri tekniğin avantajı, bileşen teleskoplar arasındaki boşluğa eşit bir açıklığa sahip devasa bir teleskobun açısal çözünürlüğüne sahip görüntü üretebilmesidir. Açıklık sentezi adı verilen matematiksel bir sinyal işleme tekniği, yüksek çözünürlüklü görüntüler oluşturma amacıyla ayrı sinyalleri birleştirmek için interferometri tekniğinde kullanılır (Estalella, 2008).

İnterferometrinin ilk kullanımı, yıldızların çaplarını ölçmek için Mount Wilson Gözlemevi'nin reflektör teleskobu üzerindeki Michelson yıldız interferometresi tarafından gerçekleştirilmiştir.

Bu nedenle astronomik interferometri esas olarak Michelson interferometreleri kullanılarak gerçekleştirilir (Hutter, 2014). Kızıl Dev Betelgeuse yıldızı, 13 Aralık 1920'de bu teknikle çapı belirlenmiş ilk yıldız olmuştur. 1940'larda ise ilk yüksek çözünürlüklü radyo astronomi gözlemlerini yapmak için radyo interferometrisi kullanılmıştır. Sonraki 30 yıl boyunca radyo dalga boylarında yapılan araştırmalar, VLT, VLA, NPOI, CHARA, MROI, ALMA ve NOEMA gibi büyük cihazların geliştirilmesine ön ayak olmuştur. İnterferometri tekniğini kullanarak geliştirilmiş bu cihazların çalışma mantığı şu şekildedir:

İnterferometre, elektromanyetik dalgaların üst üste getirilerek bilgi çıkarıldığı, astronomi, fiber optik, mühendislik metrolojisi, optik metroloji, oşinografi, sismoloji, spektroskopi, kuantum mekaniği, nükleer ve parçacık fiziği, plazma fiziği, uzaktan algılama, biyomoleküler etkileşimler konularında kullanılan önemli bir araştırma tekniğidir (Bunch ve Hellemans, 2004). Bu sistemde yani istediğimiz optik aynayı sentezleyebilmek için ışığın birçok küçük ayna üzerinden yansıtılması sağlanmalıdır. İnterferometri tekniğinin en büyük zorluğu da ışınların ilerleyişini kontrol edip büyük bir kesinlikle sentezlenen tek bir aynaya ulaşmalarını sağlamaktır. Bu nedenle yıldızdan çıkan ışığın ışınları birleştirdiğimiz aletin içine gelene kadar yol ve alışı hızının aynı olması çok önemlidir. Teleskoplar dağın değişik bölgelerinde ve değişik yükseklikte bulunurlar ve dünya sürekli döndüğü için ışığın ilerlediği yol sürekli değişir. Bu nedenle aletin içindeki yolları gecikme hattı denilen bir teknikle sürekli değiştirmek gerekmektedir ve teleskoplardan gelen ışığın aynı anda odaya ulaşması gerekmektedir. Bu durumda zamanlama çok önemlidir. Işığın hızı ya da yol aldığı süre kontrol edilemez. Ama izlediği yolların uzunlukları sürekli değiştirebilir. İnterferometri ile görüntü oluşturabilmek için kaç tane teleskop kullanıldığı önemlidir. Daha çok teleskop, elde edilecek görüntünün daha çok detaylandırılmasını sağlar. Sonuç olarak istenilen detay seviyesine ulaşmak için dünyanın her metrekaresine bir teleskop koymak mümkün değildir. Ama gerektiği kadar teleskop kullanılırsa hiçbir yerde bulanamayan ya da hiçbir şekilde elde edilemeyen verilere ulaşabilmek mümkün olabilir (Academy, 2015).

2.3. ASTRONOMİK VERİLERİ MODELLEYEN YAZILIMLAR

Günümüzde, sürekli olarak, Plateau de Bure (Guilloteau ve diğ., 1999), CARMA (Mundy, 2013), SMA (Gurwell, 2020), VLA (NRAO, 2009) ve ALMA (Schieven, 2015) gibi gelişmiş gözlem araçlarından elde edilen verilerin boyutu gün geçtikçe artmaktadır. Gelecekte daha geniş bantlı gözlem araçları ile gözlem yapabilmek imkânı olacağı ve elde edilen verilerin daha

da artacağı düşünülmektedir. Bu durum, astronomi alanının bilgisayar gücüne olan gereksinimini gün geçtikçe artırmaktadır. Astronomi aynı zamanda gelişmiş gözlem araçlarından elde edilen verilerin boyutunu azaltmaya ve bu verilerin sadece işlenebilir olanlarını almaya da ihtiyaç duymaktadır. Bu büyük boyutta verilerin modellenmesi için IRAF (Dickinson ve Valdes, 2009), AIPS (Edgemont, 2020), CLASS (Bardeau ve diğ., 2006), GILDAS (Bardeau ve Pety, 2008), XCLASS (Möller ve diğ., 2015) gibi yazılımlar geliştirilmiştir.

2.3.1. Görüntü Küçültme ve Analiz Aracı (IRAF)

IRAF, NOAO tarafından geliştirilmiş, astronomik görüntülerden piksel dizisi formatında elde edilen verilerin azaltılmasına yönelik fonksiyonlar içeren bir yazılımdır. Elde edilen veriler CCD'ler gibi görüntüleme detektörlerinden alınır. UNIX benzeri işletim sistemleri için yazılmış olmasına rağmen, Cygwin tarafından geliştirilerek tarafından Microsoft WINDOWS üzerinde kullanımı mümkün hale gelmiştir. Zamanla tüm işletim sistemleri için uyumlu ve tüm bilgisayar türleri üzerinde çalışabilir hale gelmiştir. LINUX'un gökbilimciler için geliştirilen Distro Astro işletim sisteminde varsayılan olarak kurulu halde bulunur.

IRAF komutları (aynı zamanda fonksiyon olarak da adlandırılır.) NOAO ve dışarıdan yazılımcılar tarafından geliştirilmiş paketler halinde düzenlenmiştir. Bu nedenle paketler üzerinde değişiklik yapılması veya sonradan ek paketler dahil edilmesi bakımından uygun bir yazılımdır. Astronomik nesnelerin konumlarının kalibrasyonunun, detektör pikseller arasındaki hassasiyetinin, çok sayıda görüntülerin kombinasyonunun, spektrumdaki kırmızıya veya maviye kaymalarının ölçülmesi işlemlerini gerçekleştirir. NOAO, IRAF'ın v2.16 sürümünü tüm lisans kısıtlamalarını kaldırarak yayınlamıştır. (Dickinson ve Valdes, 2009)

2.3.2. Astronomik Görüntü İşleme Sistemi (AIPS)

AIPS, radyo teleskoplardan elde edilen alınan verilerin azaltılmasını ve analiz edilmesini destekleyen bir yazılım paketidir. AIPS'in geliştirilmesi VLA'nın faaliyete geçmesinden iki yıl önce 1978 yılında NRAO tarafından başlatılmıştır. Ağırlıklı olarak, henüz yapım aşamasında olan VLA radyo teleskoplarından elde edilen verilerin işlenmesi amacı ile geliştirilmiştir. Daha sonra genel olarak kullanılan bir yazılım olması amacı ile standart veri azaltma paketi haline gelmiştir. Günümüzde hala gelişmeye ve yaygın olarak kullanılmaya devam etmektedir. LINUX ve Mac OS kullanıcıları için sürümleri bulunmaktadır. AIPS, POPS adı verilen bir

komut satırı yorumlayıcısı kullanılarak etkileşimli olarak girilen komutlarla WINDOWS işletim sistemi üzerinde de çalışmaktadır. Yaşına ve sınırlı kaynaklarına rağmen, AIPS yaygın olarak kullanılmaktadır ve hala aktif geliştirme aşamasındadır. AIPS ücretsiz bir yazılımdır ve Genel Kamu Lisansı (GNU) hükümleri kapsamındadır.

AIPS'te komutlar, karmaşık prosedürlerin tekrarlanmasını çok daha kolay hale getiren ve veri azaltma ardışık düzenleri oluşturmak için kullanılabilen bir metin dosyasına da yerleştirilebilir. FORTRAN programlama dili kullanılarak geliştirilmiş olsa da daha sonra PYTHON tabanlı bir ara yüz eklenmiştir. (Edgemont, 2020)

2.3.3. Sürekli ve Çizgi Analizi Tek Çanaklı Yazılım (CLASS)

CLASS, tek çanak teleskopta elde edilen spektroskopik verileri azaltmak amacı ile kullanılan IRAM (Institute for radio astronomy Millimeter) tarafından geliştirilmiş bir yazılım paketidir. Ayrıca süreklilik kaymalarını azaltmak için işaretleme veya odaklanma gibi temel işlevlere sahiptir. Halihazırda kullanımda olan benzer sistemlere göre CLASS'ın orijinalliği, bir gözlem tanımlanabilmesidir. Herhangi bir gözleme yönlendirme yapmak için bir numara ve veri tabanı yönetim sisteminde olduğu gibi "Seçim Kriterleri" atanır. Standart bir kurulum ile, CLASS yazılımı sadece class yazılarak başlatılabilir. CLASS, "Languages" adı verilen bağımsız işlevlere sahip farklı bölümlere ayrılmıştır:

CLASS, veri yapısını işlemek, spektrumları veya sapmaları çizmek ve kalibre etmek için tüm genel yardımcı fonksiyonları içerir. ANALYZE, kalibre edilmiş spektrumları daha ayrıntılı analiz eden işlevler içerir. FIT, spektrumda eğri uydurma işlevini yerine getirir. Ek olarak, CLASS, GILDAS adlı yazılım paketinde tanımlanan birçok işleve de sahiptir. Komut satırı yorumlayıcısı SIC, GUI ve VECTOR dilleri aracılığıyla dahil edilmiştir. GTVL, GREG1, GREG2 ve GREG3 dilleri aracılığıyla çizim işlevlerini yerine getirmektedir. ASTRO dili aracılığıyla atmosferik günlük kayıtları alınabilmektedir. CLASS, FORTRAN90 programlama dili kullanılarak yazılmıştır ve UNIX ve Mac OS platformları üzerinde çalışır. Ve XCLASS, GILDAS gibi yazılım paketlerinin oluşmasına katkı sağlamıştır. (Bardeau ve diğ., 2006)

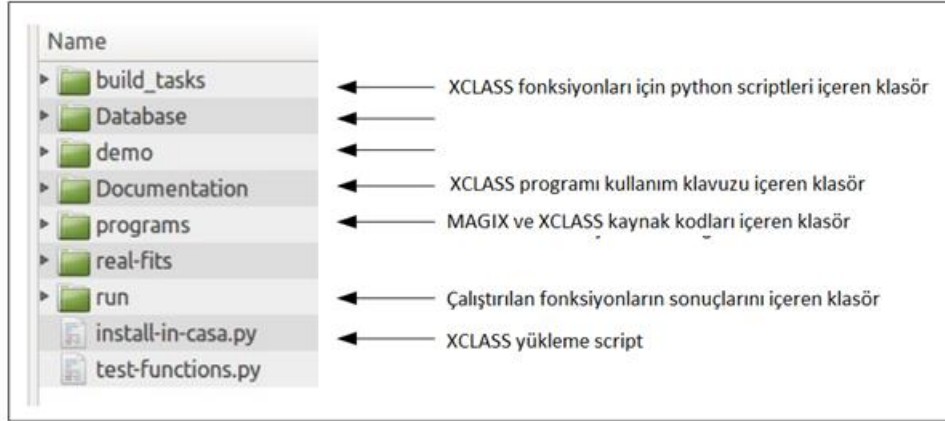
2.3.4. Grenoble Görüntü ve Çizgi Veri Analizi Yazılımı (GILDAS)

GILDAS kısa bir süre önce Groupe d'Astrophysique de Grenoble'da (şimdi LAOG, Laboratoire d'Astronomie de l'Observatoire de Grenoble) ortaya çıkan ve IRAM standart veri azaltma paketi olarak benimsenen astronomi alanında kullanılan sayısız görüntü işleme sistemlerinden

biridir. Şimdi ise IRAM & LAB (Laboratoire d'Astrophysique de Bordeaux) tarafından ortaklaşa yönetilmektedir. GILDAS, özellikle 3 boyutlu veriye yönelik her türlü spektral çizgi haritalama birçok fonksiyon içermektedir. GILDAS, kullanıcıların ihtiyaçlarını karşılamak amacı ile önceden var olan iki program olan CLASS ve GREG adında iki yazılımın üzerine eklentiler yapılarak geliştirilmesi ile ortaya çıkmıştır. VMS ortamından ortaya çıkan GILDAS, UNIX ve WINDOWS işletim sistemleri üzerinde de çalışmaktadır. (Bardeau ve Pety, 2006)

2.3.5. Geliştirilmiş CASA Çizgi Analizi Yazılım Paketi (XCLASS)

Uluslararası bilim insanlarından oluşan konsorsiyum tarafından geliştirilen CASA, ALMA ve VLA gibi gelişmiş gözlem araçlarından elde edilen ve Cologne Database for Molecular Spectroscopy (CDMS) / Jet Propulsion Laboratory (JPL) adlı veri tabanlarında depolanan spektral verileri işleme amacı ile geliştirilmiş temel işlevlere sahip yazılımdır (CASA, 2019). Sadece LINUX ve Mac OS platformları üzerinde çalışabilen XCLASS, CASA adlı sisteme entegre edilmek amacı ile Prof. Dr. Peter Schilke tarafından GILDAS ve CLASS adlı programları referans alarak geliştirilmiş bir ara yüzdür. Tek boyutlu izotermal nesneler için ışıınım transfer denklemini çözerek spektrum modelleme işlemini gerçekleştirmektedir. Şekil 2.7'de XCLASS'ın yapısı gösterilmiştir. (Möller ve diğ., 2015)

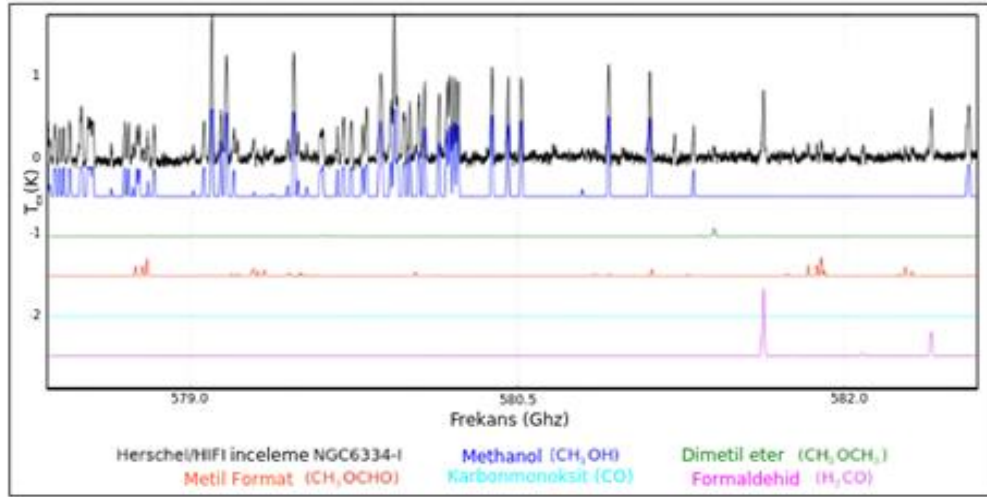


Şekil 2.7: XCLASS yazılımının yapısı. (Sánchez-Monge, 2018)

Veri tabanından veri çekme, okuma ve listeleme; veri tabanında belirli frekans aralığında geçişlere sahip verileri okuma; verilen değerlere göre grafik oluşturma işlevlerini UpdateDatabase, DatabaseQuery, ListDatabase, GetTransitions, LoadASCIIFile, myXCLASSPlot, myXCLASS, MAGIX, myXCLASSFit, myXCLASSMapFit ve myXCLASSMapRedoFit adlı fonksiyonları ile gerçekleştirmektedir (Möller ve Schilke, 2017).

2.3.5.1. Eğri Uydurma İşlemi

Her atomun, o atomun kimyasal yapısı hakkında bilgi veren ve bir nevi parmak izi özelliği taşıyan bir spektrum değeri bulunmaktadır. Örneğin Şekil 2.8’de siyah çizgi ile gösterilen çizgiler NGC6334-I adlı bulutsunun 579-582 GHz frekans aralığındaki spektrumudur. Gözlem ile elde edilen gerçek spektrum, atmosferin etkisi ve fiziksel değişimlerden dolayı nesnenin kimyasal yapısı hakkında tam bir bilgi veremez ve eğri uydurma adı verilen bir işlemle geçmesi gerekmektedir. Matematiksel olarak bir dizi veri noktasına en iyi şekilde uyan bir eğri veya matematiksel fonksiyon oluşturma işlemi olan eğri uydurma işlemi, ya verilere tam olarak uyan eğri belirler (interpolation), ya da verilere yaklaşık bir eğri uydurur (smoothing). Şekil 8’de NGC6334-I adlı bulutsunun içerisinde bulunan bazı moleküller için myXCLASSFit fonksiyonun uygulanmasından elde edilen sonuç gösterilmiştir. Böylece hangi molekülün hangi frekans aralığında katkısı olduğu görülmektedir. (Sánchez-Monge, 2018)



Şekil 2.8: NGC6334-I adlı bulutsunun 579-582 GHz frekans aralığındaki spektrumuna uygulanan eğri uydurma işlemi. (Sánchez-Monge, 2018)

2.4. PARALEL PROGRAMLAMA MODELLERİ

1970'lerin başında ortaya çıktıklarından beri sürekli gelişmekte olan mikroişlemcilerin entegre devrelerdeki transistor sayısının her iki yılda bir ikiye katlanarak artması yazılımlarda hiçbir değişiklik yapılmadan performans artışı sağlamıştır. Transistor sayısı sürekli artmaya devam etse de işlemci saat hızları 2004'ten sonra pek bir artış gösterememiştir. Bu duruma sebep olan etkenlerden biri de gerekli olan güç miktarındaki artıştır. Bir noktadan sonra saat hızında yapılan ufak artışlar bile işlemci için gereken gücü büyük miktarda arttırabilmektedir. Donanım

endüstrisi bu problemle başa çıkabilmek ve performansı makul koşullarla artırabilmek için bir entegre üzerine birden fazla işlem birimi koymaya başlamıştır. Donanımlara getirilen bu çoklu-çekirdek yaklaşımı beraberinde yazılımların tekrar değerlendirilmesi ihtiyacını getirmiştir. Çünkü bu yaklaşımda performans artışı sağlayabilmek için yazılımların bu mimariye uygun tasarlanmaları gerekmektedir. Ortaya çıkan birden fazla işlem birimini ortaklaşa ve verimli bir şekilde kullanma ihtiyacı önceleri oldukça dar bir alan olarak görülen paralel programlamayı bir anda önemli hale getirmiştir. Paralel programlama geliştirmeye yönelik bu yenilik "eşzamanlılık devrimi" olarak adlandırılmıştır. Hatta bir zamanlar neredeyse paralel programlama Yüksek Performanslı Bilgi İşlem topluluğunda yeni bir yıldız rolünü üstlenmiştir.

Bu anlamda, yazılımları paralelleştirmek için performans ve paralelleştirme kolaylığı açısından farklılık gösteren otoparallelleştirme ve paralel programlama adında iki ana yaklaşım vardır. İlk durumda, sıralı programlar, komut seviyesi paralelliği (ILP) veya paralel derleyiciler kullanılarak otomatik olarak paralelleştirme gerçekleştirilir. Mevcut uygulamaların değişiklik yapılmadan paralel bir derleyici ile yeniden derlenmesi çok büyük bir kolaylıktır. Fakat elde edilen performans kazancının beklenenden oldukça düşük olduğu gözlenmiştir. Paralel programlama yaklaşımında ise, uygulamalar paralel programlama modeli örnek alınarak iş yükünün görevlere bölünmesini ve görevlerin işlemcilerle yüklenmesi mantığı ile özel olarak geliştirilir.

Paralel bir sistem oluşturmak için birkaç işlemcinin bir araya getirilmesinin farklı yollarının olması nedeniyle paralel programlama alanında birçok olası model ortaya çıkmıştır. Yeni işlemci mimarileri, çok çekirdekli CPU'lar ve GPU'lar, heterojen paralel programlama (HPP) modelleri olarak sıralanabilir. HPP, çok çekirdekli CPU'ların ve çok çekirdekli GPU'ların yeteneklerinden yararlanmaya çalışır. (Ergün ve Sayar, 2014)

2.4.1. POSIX Thread

Taşınabilir İşletim Sistemi Arayüzü (Portable Operating System Interface), Pthreads olarak da adlandırılan bu model, birbirinden bağımsız olarak kontrol edilebilen eş zamanlı iş parçacıklarına sahiptir. İçerdiği kütüphane, iş parçacıkları oluşturmak, sonlandırmak ve seçilen bellek alanına özel erişim sağlamak için tasarlanmış iş parçacıklarını koordine eden işlemlere sahiptir. Bu modelde, yığın bellek, dinamik olarak oluşturulmuş global değişkenler ve iş parçacıkları tarafından paylaşılır. Bu özellik programlamada birtakım zorluklara neden olur.

Örneğin, birden fazla iş parçacığı aynı global değişkene erişmek istediğinde kilitlenme durumları meydana gelir. Pthreads bu durumu önlemek amacı ile kod, yalnızca bir iş parçacığının global değişkene ulaşması gereken bölümünü korumak için, muteks (karşılıklı dışlama) ve semaforlar oluşturur. Muteks, tek seferde yalnızca bir iş parçacığının kritik bölüme erişmesine izin verirken, semaforlar birkaç iş parçacığının kritik bölüme erişmesine izin verir. Yine de Pthreads genel amaçlı bir paralel program geliştirme teknolojisi olarak önerilmemektedir. Özel durumlarda ve uzman programcılar tarafından hala kullanılan bir model olsa da Pthreads'in yapılandırılmamış yapısı, doğru ve sürdürülebilir yazılımların geliştirilmesini zorlaştırır. Ayrıca, iş parçacığı sayısı, işlemci sayısı ile bağlantılı olmadığından, Pthreads programları çok sayıda işlemci ile kolayca ölçeklendirilemez. Tüm bu nedenlerden dolayı POSIX, yüksek performans gerektiren uygulamalar için uygun değildir. (Diaz ve diğ., 2012)

2.4.2. OpenMP

OpenMP, amacı paralel programlama modelini kolaylaştırmak olan, paylaşımlı bellek platformunda çalışan, fonksiyon odaklı, uygulama programlama ara yüzüdür- application programming interface (API). OpenMP çoklu iş parçacıklı ara yüzü, yüksek performans gerektiren hesaplamalar için tasarlanmıştır. OpenMP Pthreads'lerden birkaç önemli noktada farklılık göstermektedir. Bunlardan birincisi Pthread tamamı ile bir kütüphane olarak uygulanırken; OpenMP, hem iş parçacığı havuzunun yönetimini sağlayan bir dizi derleyici yönergesi, pragma hem de kütüphane olarak uygulanır. Yönergeler derleyiciye iş parçacıkları oluşturması, senkronizasyonunun gerçekleştirmesi ve paylaşılan bellek alanının yönetilmesi talimatını verir. Bu nedenle, OpenMP'nin bu yönergeleri anlamak ve işlemek için özel bir derleyiciye ihtiyacı vardır. Şu anda, Fortran, C ve C++ programlama dilleri için gittikçe artan sayıda derleyiciler mevcuttur.

OpenMP, paralel uygulamalar için, kodun sıralı ve paralel bölümleri arasındaki geçişte çatal/birleşim (fork/join) modeli uygulanır. Çatal olarak adlandırılan durumda, tek bir iş parçacığı birden fazla bağımsız iş parçacıklarına bölünür. Tüm iş parçacıkları belirtilen görevleri tamamladığında, sıralı yürütmeye (birleştirme) işlemine devam edilir. OpenMP, ana iş parçacığı (sırasıyla yürütülen komutlar dizisi) belirli bir sayıda yardımcı iş parçacıklarını durdurur ve bir komutu yardımcı iş parçacıkları arasında paylaştırır. Yardımcı iş parçacıkları birbiri ardında paralel şekilde çalışırlar. Farklı işlemcilerin iş parçacıkları farklı çalışma zamanı

ortamlarını kendilerine tahsis ederler. Paralel çalışacak olan kodun bir bölümü sırasıyla işaretlenir. Bu işaretleme kod bölümünün yürütülmesinden önce iş parçacıklarının o kod bölümüne girmelerini sağlayan ön işlemci direktifleridir. Program sonlanana kadar bu işlem devam eder. OpenMP'nin eksik tarafı ise, senkronizasyon hatalarıyla karşılaşma riskinin çok fazla olmasıdır. (Diaz ve diğ., 2012)

2.4.3. Mesaj İletim Arayüzü (MPI)

MPI, işlemciler arasında mesaj aktarım yolu ile iletişimin sağlandığı paralel programlama modeli kullanan bir kütüphanedir. Programlama için kullanılacak olan derleyicilerin MPI kütüphanesi ile bağlantısı olmalıdır. MPI 'ın diğer mesaj aktarım yolu ile işlemciler arasında iletişimi sağlayan eski kütüphanelere olan üstünlüğü taşınabilir ve hızlı olmasıdır.

MPI, işlemciler arasındaki iletişimi mesaj aktarım yolu ile sağlayan paralel programlama modelidir. Bu modelde paralel olarak yürütülen işlemler, ayrı bellek alanlarında gerçekleştirilir. İletişim, bir işlemin adres alanının bir kısmı başka bir işlemin adres alanına kopyalanması ile sağlanır. Bu iletişim birinci adres alanındaki işlem mesaj gönderimi, ikinci adres alanındaki işlem mesaj alma işlemini yürüttüğünde gerçekleşir. MPI'da hangi işlemcinin hangi işlemi yürüteceği esasen işlemciler arasındaki görev dağılımı programcı tarafından belirlenir. MPI'da noktadan noktaya, toplu, tek taraflı ve paralel giriş/çıkış olmak üzere dört çeşit iletişim modeli bulunmaktadır. MPI'daki iletişim modelleri, noktadan noktaya, toplu, tek taraflı ve paralel G/Ç işlemlerinden oluşur. Mesajların gönderimi ve alımı sırasında, ilişkili ara bellek alanını yönetmek için bir miktar senkronizasyon gereklidir. (Diaz ve diğ., 2012)

2.4.4. OpenCL

OpenCL, CPU'lar, GPU'lar, FPGA ve diğer işlemciler arasında genel amaçlı paralel programlama modeli için telif hakkı olmayan bir standarttır. OpenCL, AMD, Intel, NVIDIA ve ARM tarafından desteklenmektedir. OpenCL'nin ilk versiyonu, OpenCL 1.0, 2008 sonlarında Khronos Group tarafından piyasaya sürülmüştür. OpenCL'in esas amacı, çekirdek (OpenCL cihazlarında çalışan işlevlere verilen isim) oluşturmak ve oluşturulan bu çekirdekleri yönetmek için API'ler yazmaktır. Çekirdekler, programın çalışması sırasında hedef cihaz için özel olarak derlenir. Bu durum, sistemdeki tüm cihazlardan maksimum performans elde edilmesini sağlar.

Her OpenCL cihazı bir dizi hesaplama biriminden; her hesaplama birimi ise bir dizi iş-parçacıklarından oluşur. Çekirdek, iş-parçacıkları olarak adlandırılan ve benzersiz bir kimliği

sahip çalıştırılabilir temel kod birimidir. İş-parçacıkları, iletişim ve senkronizasyonu sağlamak için çalışma grupları halinde organize edilebilir. Bir OpenCL programının yürütülmesi, bir çekirdeğin, bir veya daha fazla sayıda OpenCL cihazında aynı anda yürütülmesini anlamına gelir. Bellek alanı tahsis etme, bellekler arası veri transferi ve komutların çalıştırılma sırasını kontrol etmek için kullanılan kuyruklarının oluşturulması dahil programdaki her işlem CPU tarafından kontrol edilerek sıraya alınır. Fakat, komutları senkronize edilmesi işlemi programcı tarafından gerçekleştirilir.

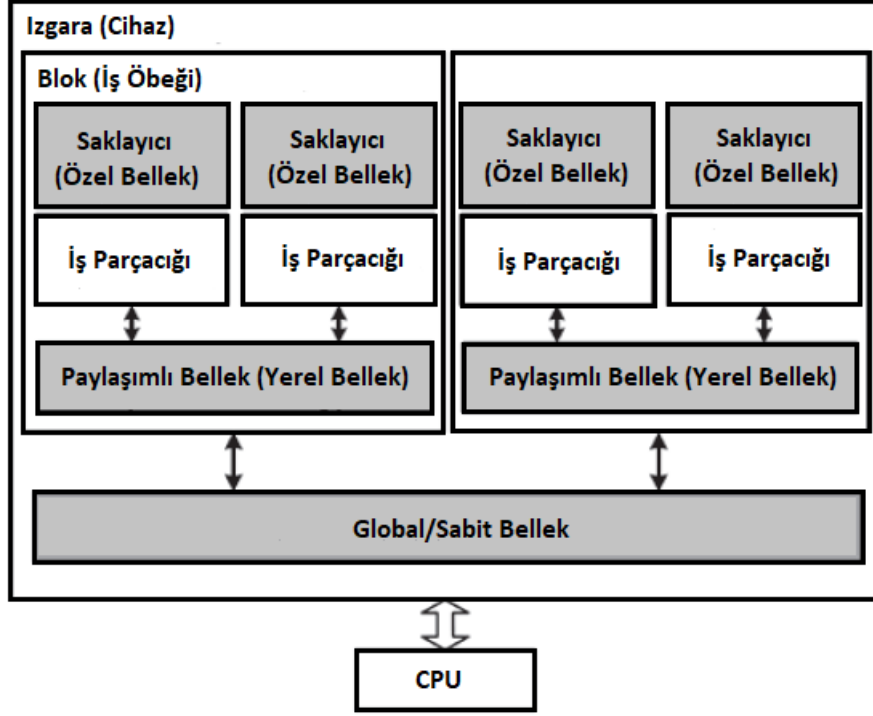
OpenCL, çok sayıda bellek modeline sahiptir. Bunlardan birincisi, tek bir iş-parçacığı tarafından kontrol edilen özel bellek alanıdır. İkincisi, bir çalışma grubundaki tüm iş-parçacıkları tarafından kullanılabilen yerel bellektir. Üçüncüsü, tek bir OpenCL cihazında bulunan bir iş-öbeğinin iş-parçacıklarının salt-okunur erişim için ayrılan sabit bellek alanıdır. Sabit bellek, CPU tarafından yazılır ve okunur ancak bir çekirdeğin çalışma esnasında salt-okunur durumdadır. Son olarak, OpenCL cihazında bulunan tüm çalışma gruplarının tüm iş-parçacıkları tarafından okunabilen ve yazılabilen global bellek alanıdır. (Diaz ve diğ., 2012)

2.4.5. CUDA

2003 yılında San Diego’da düzenlenen Siggraph/Eurographics Graphics Hardware isimli çalıştayda, GPGPU kavramı ortaya çıkmıştır. Bugün bir sistemde bir veya daha fazla sayıda CPU veya GPU bulunabilir ve bu tür sistemlere heterojen sistemler denir. Heterojen sistemin amacı, mikroişlemciler endüstrisinde Accelerated Processing Units (APU-Hızlandırılmış İşlemci Birimleri) geliştirilmektir. APU, CPU (çok çekirdekli) ve bir GPU’yu aynı kalıpta birleştiren bir modeldir ve bu model ile daha iyi bir veri aktarım hızı ve daha düşük güç tüketimi sağlanmıştır. AMD Fusion ve INTEL Sandy Bridge, APU’ların bu özelliğe sahip modellerine verilebilecek en iyi örneklerdir. CPU+GPU sistemlerde ilk olarak BROOK ve CG gibi programlama dilleri kullanılmıştır. Daha sonra NVIDIA firması, 2007 yılının başlarında bir model ve programlama dili olarak Compute Unified Device Architecture ‘ı (CUDA) popüler hale getirmiştir. Daha sonra C, C++, FORTRAN, JAVA, PYTHON ve MATLAB programlama dillerine eklenti yapılarak CUDA C, CUDA Fortran gibi programlama dilleri oluşturulmuştur.

CUDA modeli, sürekli artan sayıda transistör ve işlemci sayısı ile ölçeklenebilir yazılımlar geliştirmek için tasarlanmıştır. Günümüzde kullanılan işlemciler en fazla 18 çekirdeğe sahipken NVIDIA Tesla K20X adlı ekran kartında 2688 adet çekirdek bulunmaktadır. CUDA’da paralel

bir sistem, bir ana bilgisayar (CPU) ve bir hesaplama kaynağı veya cihazından (GPU) oluşur. İşlevlerin GPU'da paralel olarak çalışması bir dizi iş parçacığı (thread) tarafından gerçekleştirilir. GPU, bir veya birden fazla ızgara içerisinde bir veya birden fazla bloğu bulunduran işlemci koleksiyonudur. GPU iş parçacığı mimarisi, Şekil 2.9'da gösterildiği gibi blok ve ızgara olmak üzere iki seviyeli bir hiyerarşiden oluşur. Bloklar, iş parçacığı kimliği ile tanımlanan bir dizi iş parçacıklarından oluşur. Izgaralar ise aynı boyuta sahip bir dizi bloklardan oluşan yapıya sahiptir. Bloklar arasından herhangi bir senkronizasyon yoktur ve tüm ızgara tek bir GPU tarafından yönetilir. Bir blok asla birden fazla işlemci üzerinde bulunamaz. Blok içindeki iş parçacıkları, paylaşılan bellek alanında veri paylaşarak ve bellek erişimlerini koordine etmek için çekirdekleri senkronize ederek iş birliği yapabilir. CUDA'da programcı iş parçacığı oluşturma ve yok etme işlemlerinden sorumlu değildir. Programcı sadece bir görevi yerine getirmek için gereken ızgara ve blok boyutunu belirlemesi gerekir. Bu neden CUDA'da iş yönetimi dolaylı olarak gerçekleşir. Ayrıca CUDA'da iş yükü paylaşımı ve adresleme açıkça yapılır. Programcı, her ızgaranın ve bloğun boyutunu belirleyerek ve "Global Function" ile belirtilen çekirdekler ile paralel olarak çalıştırılan iş yükünü tanımlanmış olur.



Şekil 2.9: CUDA mimarisi ve bellek modeli. (Diaz ve diğ., 2012)

Şekil 2.9’da belirtilen CUDA bellek modelinde global ve sabit olmak üzere iki çeşit bellek türü görülmektedir. Bu global ve sabit bellek türleri, CPU üzerinde çalışan kod tarafından, okuma ve yazma amacı ile erişilebilir. Ancak sabit belleğin GPU tarafından salt-okunur erişim izni vardır. Bir blok içerisinde ise, paylaşılabılır bellek ve saklayıcı olarak da bilinen yerel bellek bulunur. Paylaşılabılır bellek alanına blok içerisindeki tüm iş parçacıkları tarafından erişim izni bulunmaktadır. Saklayıcılar ise her iş parçacığı için bağımsız olarak bulunur.

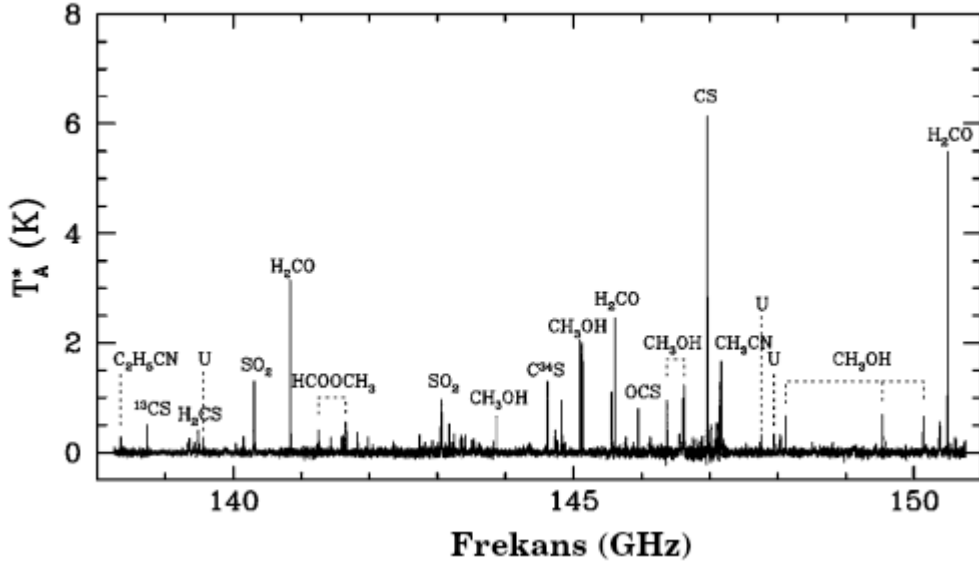
Son olarak, INTEL firması yakın zamanda Knights Ferry adında bir girişimde bulunmuştur. Bu girişimi INTEL Many Integrated Core (MIC) mimarisi ile geliştirilmektedir. Knight Ferry, 32, x86 mikroişlemciler ile, bir PCI kart üzerine uygulanmaktadır. MIC, CUDA'dan daha klasik bir uyumlu paylaşılan bellek paralel programlama paradigmasını desteklemektedir. INTEL’in yerel C/C++ derleyicileri kullanılarak programlanacaktır. (Diaz ve diğ., 2012)

2.5. OTOMATİK ÇİZGİ TANIMA YAZILIMI

Gökyüzünden alınan verilerden elde edilen spektrumun içinde bulunan her bir Gauss eğrisinin varlığı ile yükseklik, frekans ve genlik gibi parametrelerinin tespiti; incelenen gökyüzü bölgesindeki moleküllerin belirlenmesi, bölgenin yapısının çıkarılması ve bu bölgenin diğer

bölgeler ve dünya üzerindeki etkilerinin belirlenmesinde büyük önem arz etmektedir. Her atom ve molekülün bulunduğu ortamın fiziksel koşullarına bağlı olarak kendine özgü emisyon ve soğurma çizgileri vardır. Bu sebeple, gök cisimlerinin spektrumlarındaki söz konusu spektral çizgilerin incelenmesi sayesinde, gök cisimlerinin fiziksel özellikleri anlaşılabilir (Uslu, 2019). Bu nedenle, spektrum analizi, herhangi bir nesnenin fiziksel özellikleri hakkında bilgi edinebilmek için kullanılan en önemli yöntemlerden biridir.

Atom veya molekül yapısı içindeki elektronların enerji seviyelerindeki değiřtirmelerine genel olarak “geçiř” adı verilir ve bu geçiřler atom veya moleküllerin fotonları soğurmaları sonucunda meydana geldiğinden- Kirschhoff yasaları ile tanımlandığı üzere- spektrum çizgileri (sürekli spektrum üzerinde soğurma çizgileri veya karanlık zeminde emisyon çizgileri) oluşur. Belli geçiřler ancak belli enerjiye sahip fotonların soğrulması ile olur. Elektron daha sonra eski seviyesine geri döner ve aynı enerjiye sahip bir foton yayınlar. Bazen birkaç farklı seviyeden adım adım geri döner ve her seferinde belli enerjilere sahip fotonlar yayınlar. Bunlar Gauss eğrilerinin genişlemesine katkıda bulunurlar. Laboratuvarıda üretilen atom ya da molekül çizgileri durgun (rest) haldedir, diğeri bir deyiřle hızları yoktur. Ama astronomik bir ortamdan kaynaklanıyorsa, bir hızları olur ve bu durum çizgilerde dalga boyu kayması ve çizgi genişlemesi olarak görülür. Bütün bunların hepsi ve daha fazlası birleřince molekül bantları oluşur. Şekil 2.10’da gösterildiğı gibi her atomun ve molekülün kendine özgü spektrumu olduğundan, bir gökyüzü bölgesinin spektrumunun özelliklerinin ortaya konması o bölgenin radyal hızı, sıcaklığı, molekül miktarı gibi birçok fiziksel bilgi elde edilmesinde önemli rol oynar. (Tennyson, 2005)



Şekil 2.10: Orion KL adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektrumdaki moleküler bantlar. (Tennyson, 2005)

Güneş sisteminin dışındaki astronomik nesneler hakkındaki tüm bilgiler, bize ulaşan elektromanyetik radyasyonun (ışık) incelenmesi ile elde edilir. Bu ışık, dikkatli bir analizle çok ayrıntılı bilgiler verir. Gözlem yapmak için kullanılan cihazların farklı dalga boylarına duyarlılık derecesine spektral çözünürlük denir. Gözlemlenerek elde edilen ışık, spektral çözünürlüğe göre, genel kategoriler kullanılarak sınıflandırılabilir.

Gece gökyüzüne çıplak gözle bakıldığında çoğu astronomik cisim beyaz görünür. Beyaz ışık aslında farklı renklerine ayrıştırılmayan birçok dalga boyundan oluşan ışıktır. Beyaz ışık, gece gökyüzündeki nesnelerin konumlarını verir ve yıldızların ve galaksilerin haritalarını oluşturmak için kullanılabilir. Kuyruklu yıldızlar gibi gök cisimlerinin gece gökyüzünde hareketlerini çizmek için de kullanılabilir.

Mars ve Jüpiter gibi bazı gök cisimlerine veya Betelgeuse gibi yıldızlara dikkatle bakıldığında, bu nesnelerin belirli bir renkte olduğu görülebilir. Düşük çözünürlüğe sahip aletler kullanarak, Dünya'ya gelen ışığı renk bantlarına ayırmak mümkündür. Renkleri gözlemleyerek sıcaklıklar hakkında bir şeyler öğrenebiliriz. Örneğin, mavi yıldızlar kırmızı olanlardan daha sıcaktır; Güneş koronası gibi X ışınları yayan nesneler çok sıcaktır, soğuk nesneler ise yalnızca radyo dalgaları gibi çok uzun dalga boylarında ışık yayabilir.

En ayrıntılı astrofiziksel bilgiler, yalnızca, dünyaya gelen, çeşitli dalga boylarının birleşiminden oluşan ışığın tespit edilmesini içeren yüksek çözünürlüklü çalışmalar ile elde edilir. En yüksek

çözünürlük ile gerçekleştirilen bu tür çalışmalar yalnızca herhangi bir cismin merkezi dalga boyunu değil, aynı zamanda şeklini de verir. Bir spektrumu yorumlamak için, atomik ve moleküler fizik hakkında bilgiye ihtiyaç vardır. Temel fiziksel parametrelerden oluşan bu bilgi, genellikle laboratuvar çalışmalarından elde edilir. Bu fiziksel parametreler ile spektrum arasında doğrudan bir ilişki vardır. Bu nedenle, bir spektrumda bulunan herhangi bir çizgiden, aşağıdaki bilgileri çıkarmak için laboratuvar verileri kullanılır.

Bileşim: Gözlemlenen cismin bileşimi, laboratuvar ortamında hangi atomun, iyonun veya molekülün gözlemlenen geçişi (spektral çizgi) oluşturduğu bilinmesi ile elde edilir.

Sıcaklık: Sıcaklık, gözlemlenen cismin atomunun enerji seviyeleri arasında elektronların yer değiştirmesi ile oluşur. Herhangi bir atomun farklı enerji seviyeleri arasında çok sayıda geçişler meydana gelir. Hangi enerji seviyelerinde geçişler olduğunu bilmek, gözlem yapılan cismin ortalama sıcaklığı hakkında doğrudan bilgi verir.

Yoğunluk: Herhangi bir atomun enerji seviyeleri arasında gerçekleşen geçişlerin yoğunluk değeri, ancak gözlemlenen geçişin şiddeti biliniyorsa belirlenebilir. Geçişlerin şiddet değerinin laboratuvar ortamında belirlenmesi zordur. Astronomik olarak, bir geçişin şiddeti, optik derinlik değerinin uygun koşulları altında geçişe sahip atomların sayısı ile doğrudan ilişkilidir. Bu nedenle, geçişlerin yoğunluğunun bilinmesi, herhangi bir atomun yoğunluğunu belirlemek için önemlidir.

Hareket: Cisimlerin veya cismi içeren bölgenin tamamının dünyaya göre gözlemlenen hareketleri, spektral verilerde, yani dalga boyunda bir kaymaya yol açar; bu kayma Doppler kayması olarak bilinir. Doppler kayması, ışığın dalga boyunun laboratuvar ortamında ölçülen konumunda meydana gelen değişiktir. Bu kayma değeri denklem 2.1’de belirtilen Doppler formülüyle hesaplanır.

$$\frac{v}{c} = \frac{\Delta\lambda}{\lambda} \quad (2.1)$$

Denklem 2.1’de, v değeri kaynağın bize göre hızını, $c=2.99792458 \times 10^8 \text{ m.s}^{-1}$ değeri ışık hızını, λ laboratuvar ortamında durağan halde oluşan geçişin dalga boyu değerini, $\Delta\lambda$ ise dalga boyu değerindeki değişimi (Doppler kayma değeri olarak da bilinir) gösterir. Bu formülün uygulanması için, dalga boyunun yüksek doğrulukta laboratuvar ortamında ölçülen değerleri gerekir ve göreceli olmayan hareketlerin belirlenmesinde kullanılır. Bir cisim bize doğru

hareket ettiğinde, geçiş daha kısa dalga boylarına ("maviye") veya nesne bizden uzaklaştığında, daha uzun dalga boylarına ("kırmızıya") kayar. Edwin Hubble'ın (1889-1953) 1929'da evrenimizin Büyük Patlamadan sonra tek bir noktadan tekdüze bir şekilde genişlediğini göstermesine izin veren şey, hidrojen atomlarının spektrumlarında meydana gelen Doppler kaymasını keşfetmesidir.

Basınç: Spektral verilerdeki çizgi profili izlenerek ortamın basıncı veya yoğunluğu elde edilir. Spektral çizgiler, atomlar arasındaki çarpışma sayısı arttıkça genişler; bu çarpışmalar ne kadar sık olursa genişleme o kadar büyük olur. Bu süreç "basınç genişlemesi" olarak da adlandırılır. Ayrıca Doppler formülünde belirtildiği gibi termal hareketler spektral çizgileri genişletir. Doppler genişlemesi, sıcaklığı fazla olan atomların soğuk olanlardan daha hızlı hareket etmesinden kaynaklanır. Bu tür gözlemler için özellikle yüksek çözünürlük gereklidir. Çünkü basınç ve sıcaklığın spektrum üzerinde oluşturduğu iç içe geçmiş çizgiler yalnızca çok yüksek çözünürlüklü gözlemler kullanılarak çözülebilir.

Manyetik Alan: Gözlem yapılan bölgede mevcut olan herhangi bir manyetik alan, birden fazla bileşene ayrılabilen spektral çizgiler üzerinde gözlemlenebilir. Açısal momentuma sahip durumların enerji seviyeleri, ancak bir manyetik alanın var olması ile bölünür. Bir spektral çizgi bölünerek iki veya daha fazla sayıda farklı iç içe geçmiş spektral çizgiler haline gelebilir. Bu bileşenler arasındaki ayrım derecesi, doğrudan gözlem yapılan bölgenin manyetik alanının gücüne bağlıdır. Böylece bu tür bölünmeler gözlemlenerek, manyetik alanın gücü belirlenir.

Bu tür gözlemlerden elde edilen bilgiler, astronomi alanında yapılan çoğu çalışmalar için anahtar nitelikte bilgilerdir. Bununla birlikte, herhangi bir gözlemden elde edilen spektrumu yorumlanması, spektrumların kendine özgü özellikleri hakkında ayrıntılı bilgi gerektirir. Gözlemlenen her atom veya iyon veya molekül için bilinmesi gerekenler:

- 1) Önemli spektral çizgiler.
- 2) Enerji seviyesi yapısı.
- 3) Gözlemlenen bölgeden elde edilen spektral verilerdeki geçişlerin kuvveti.
- 4) Gözlenen herhangi bir geçişin laboratuvar ortamında elde edilen dalga boyu değeri.

Meydana gelen basınçlı ortamın sebep olduğu spektral çizgilerde genişleme ve manyetik alan nedeni ile spektral verilerdeki bölünmeyi yorumlamak için önemli ölçüde kuantum mekaniği bilgisi gereklidir.

Bu tür özelliklerin belirlenmesi, spektrumdaki iç içe geçmiş her Gauss eğrisinin varlığının ve parametrelerinin belirlenmesi ile gerçekleşir. Bu tez çalışmasının ikinci aşamasında, ALMA ile güneş kütlelerinin yaklaşık 3 milyon katı kütleye sahip ve galaksideki en büyük ve karmaşık büyük kütleli yıldız oluşum bölgelerinden biri olan, Sagittarius B2 (SgrB2) adlı büyük kütleli yıldız oluşum bölgesinden alınan spektral verilerdeki, veri tabanında bulunan C_2H_5OH , CH_3CN , CH_3OH , $HCCCN$, SO_2 , SO molekül bantlarındaki, Gauss eğrilerinin tespiti işlemi gerçekleştirilmiştir. ALMA teleskopları, gerektiğinde 16km çaplı tek bir radyo teleskop gibi işlev gördüğünden milimetre ve milimetre altı dalga boylarında radyo sinyallerini algılayabilmektedir.



Şekil 2.11: Frekans bandına ve Radyo teleskobun çapına bağlı olarak Gauss eğrisinin değişimi. (RF Wireless World, 2017)

Şekil 2.11'de görüldüğü gibi anten çapının artması ile elde edilen spektrum bant genişliği daralmaktadır. Bu durum ALMA cihazlarından elde edilen spektrumun analizini zorlaştırmıştır. Bu çalışma sayesinde, Otomatik Çizgi Tanıma Yazılımı sadece büyük kütleli yıldız oluşum bölgeleri üzerinde değil, evrende başka bölgeler üzerinde uygulanabilecek; bu sayede, evrenin fiziksel ve kimyasal yapısı hakkında daha fazla bilgi sahibi olunmasına katkıda bulunacaktır. Bu kadar büyük ve karmaşık bir alandan elde edilen verilerin başarılı bir şekilde analiz edilmesi, evrenin farklı bölgelerinde de analiz yapılabileceği anlamına gelecektir, böylece karanlık evrenin fiziksel ve kimyasal yapısı hakkında daha fazla bilgi edinmek mümkün olacaktır. |

3. MALZEME VE YÖNTEM

3.1. XCLASS

XCLASS, spektral veri içerisinde çizgi-yoğun yerleri tanımlama işlevini gerçekleştiren, bir spektrumu keyfi sayıda molekül ile modelleyen ve böylece her bir molekülün nerede katkısı olduğunu tanımlayabilen yazılım paketidir. Modelleme işlemini gerçekleştirirken verilerin izole bir ortamdan elde edildiği yani hiçbir enerji akışının olmadığı, LTE olduğu varsayılır. XCLASS, kullanıcı tarafından belirlenen molekül sayısı ile bir spektrumu modelleyebilir. Modelleme işleminde kullanılacak olan her molekülün bileşen sayısı yine kullanıcı tarafından belirlenir. Tek boyutlu izotermal nesneler çok basit bir geometrik yapıya sahip olduğundan modelleme işlemi gerçekleştirilirken bileşen sayısı varsayılan olarak 2 değeri atanır.

Bileşenlerden birincisi, Şekil 3.1’de bulunan ışının ortasında merkezlenmiş nesneler topluluğundan oluşan çekirdektir (spektral analiz bölümünde bahsedilen emisyon çizgilerine karşılık gelir). Çekirdek, üst üste binen, fiziksel olarak birbiri ile etkileşimde olmayan sıcak yoğun kümeler olarak adlandırılır. Toz emisyonunun (kısmen) bu çekirdek bileşenlerinden yayıldığı varsayılır. Işın doldurma faktörü ile ağırlıklandırılıp Denklem 3.2 ile yoğunlukları hesaplanır.



Şekil 3.1: Teleskoba gelen bir ışının içindeki çekirdek katmanlarının dağılımı. (Möller ve Schilke, 2017)

Şekil 3.1’de ışın ortasında ortalanmış şekilde duran 1, 2 ve 3 üç farklı çekirdek bileşeni olarak varsayılır. Farklı renklerle gösterilen bu bileşenler farklı kaynak boyutuna, uyarım sıcaklığına sahiptir. Ek olarak, tüm çekirdek bileşenlerin teleskopa aynı mesafeye sahip olduğu varsayılır, yani tüm çekirdek katmanları görüş hattına dik bir düzlem içinde yer almaktadır.

Bileşenlerden ikincisi, çekirdek bileşenlerin önündeki katmanlar halinde bulunduğu olduğu varsayılan ön plan nesnelerdir (spektral analiz bölümünde bahsedilen soğurma çizgilerine karşılık gelir). Mevcut olan tek boyutlu uygulamalarda çekirdek arka plan olarak kabul edilir ve tüm ışının ortalama yoğunluğuna sahip olur. Tüm ışın görüş hattı boyunca sırası ile kaynak, yoğun çekirdek bileşenleri ve son olarak soğurma bileşenleri şeklindedir. Şekil 3.1'de gösterildiği gibi, çekirdek bileşenlerinin birbiriyle radyasyonel olarak etkileşim içinde olmadığı varsayılır, yani bir çekirdek katmanı diğerlerinden etkilenmez. Fakat birkaç molekül ve bileşikten oluşan kaynakların olabileceği olasılığından çekirdek katmanları çakışabilir.

Ayrıca çekirdek düzlemin, Şekil 3.1'de 4 numara ile belirtilen tüm ışın boyunca homojen yoğunluğa sahip bir arka plan tabakasının önünde bulunduğunu varsayılır. Çekirdekte bulunan bileşenlerin birbirleri ile radyasyonel olarak etkileşimleri yoktur. Çekirdek katmanlar için ışınım transfer denklemi Denklem 3.1'deki gibidir.

$$T_{mb}^{core}(v) = \sum_m \sum_c [\eta(\vartheta^{m,c}) [S^{m,c}(v)(1 - e^{-\tau_{total}^{m,c}(v)}) + I_{bg}^{core}(v)(e^{-\tau_{total}^{m,c}(v)} - 1)]] + (I_{bg}^{core}(v) - I_{CMB}) \quad (3.1)$$

Denklem 3.1'deki toplam sembolü, her molekül ve bileşen için bu denklemin hesaplanacağı anlamına gelir. Denklem 3.1'deki m molekülünün ve c bileşeninin ışın doldurma (seyreltme) faktörü Denklem 3.2 ile hesaplanmaktadır. $\vartheta^{m,c}$ ve ϑ_t sırasıyla kaynak ve teleskop ışınının FWHM değerini yani y eksenini üzerinde ölçülen maksimum genliğin yarısını temsil eder. Bu değerler kullanıcı tarafından XCLASS içerisinde MOLFIT uzantılı dosyada tanımlanır.

$$\eta(\vartheta^{m,c}) = \frac{(\vartheta^{m,c})^2}{\vartheta_t(v)^2 + (\vartheta^{m,c})^2} \quad (3.2)$$

Bir teleskop ile ayırt edilebilen iki kaynağın minimum açısal ayırım değeri, gözlemlenen ışığın dalga boyuna ve teleskobun çapına bağlıdır. Bu açı, kırınım limiti (diffraction limit) olarak adlandırılır. Bu nedenle, tek cihaz ile yapılan gözlemlerde, teleskop ışınının FWHM değeri, kırınım limiti ve teleskobun çapına bağlıdır. Kırınım limit değeri Denklem 3.3 ile hesaplanır.

$$\vartheta_t(v) = (1.22 \cdot \frac{\lambda}{D}) \cdot \varepsilon = (1.22 \cdot \frac{c_{light}}{v} D) \cdot \varepsilon \quad (3.3)$$

D teleskobun çapını, c ışık hızını ve $\varepsilon = 3600.180/\pi$ değeri teleskop ışınının FWHM değerini arcsec cinsinden elde etmek için kullanılan dönüşüm faktörü değerlerini belirtir. İnterferometrik

gözlemlerden elde edilen spektral veriler için FWHM değeri kullanıcı tarafından belirtilmelidir. Bu nedenle, tüm frekans aralığı için sabit bir interferometrik ışın FWHM değeri belirlenmiştir.

Denklem 3.1'deki $\eta_{\max}^{\text{core}}$ değeri tüm molekül ve bileşenlerin en büyük ışın doldurma faktörü değeridir. Genel olarak radyasyon sıcaklığının parlaklık değeri $J(T, \nu)$ Denklem 3.4 ile hesaplanır.

$$J(T, \nu) = \frac{h\nu}{k_B} \frac{1}{e^{h\nu/kT} - 1} \quad (3.4)$$

Denklem 3.1'deki J_{CMB} değeri, kozmik mikrodalga arka planın sıcaklığını temsil eder. ($T_{\text{CMB}} = 2.7\text{K}$ için $J(T_{\text{CMB}}, \nu)$)

Denklem 3.1'deki $S^{m,c}(\nu)$ değeri, Kirchhoff'un termal radyasyon yasasına göre kaynak fonksiyonu temsil eder ve aşağıdaki Denklem 5 ile hesaplanır.

$$\begin{aligned} S^{m,c}(\nu) &= \frac{\epsilon_l^{m,c}(\nu) + \epsilon_d^{m,c}(\nu)}{K_l^{m,c}(\nu) + K_d^{m,c}(\nu)} \\ &= \frac{K_l^{m,c}(\nu)J(T_{\text{ex}}^{m,c}, \nu) + K_d^{m,c}(\nu)J(T_d^{m,c}, \nu)}{K_l^{m,c}(\nu) + K_d^{m,c}(\nu)} \\ &= (1 - \delta_{y,0}) \cdot \left[\frac{\tau_l^{m,c}(\nu)J(T_{\text{ex}}^{m,c}, \nu) + \tau_d^{m,c}(\nu)J(T_d^{m,c}, \nu)}{\tau_l^{m,c}(\nu) + \tau_d^{m,c}(\nu)} \right] + \delta_{y,0}J(T_{\text{ex}}^{m,c}, \nu) \end{aligned} \quad (3.5)$$

Denklem 3.5'de $\epsilon_{l,d}^{m,c}$ ve $\kappa_{l,d}^{m,c}$ değerleri sırasıyla çizgi ve toz için çekirdek ve ön plan katsayılarını temsil eder. Moleküllerin ve tozun karışım halinde olduğu durumlarda optik derinlik değeri $\tau^{m,c}(\nu) = \int \kappa^{m,c}(\nu) ds = \kappa^{m,c}(\nu)s$ ile hesaplanmaktadır.

Her bir molekülün ve bileşenin toplam optik derinliği $\tau_{\text{total}}^{m,c}$, her bir molekülün ve bileşenin tüm çizgilerinin optik derinlikleri $\tau_l^{m,c}$ ve toz optik derinliğinin $\tau_d^{m,c}$ toplamı olarak tanımlanır ve Denklem 3.6 ile hesaplanır.

$$\tau_{\text{total}}^{m,c}(\nu) = \tau_l^{m,c}(\nu) + \tau_d^{m,c}(\nu) \quad (3.6)$$

Toz optik derinlik değeri ise Denklem 3.7 ile elde edilir. Bu denklemde $N_H^{m,c}$ hidrojen sütun yoğunluk değerini, $\kappa_{\nu_{\text{ref}}}^{m,c}$ kullanıcı tarafından referans ν_{ref} olarak belirlenen frekans değerinde toz kütle opaklığı değerini, $\beta^{m,c}$ ise $\kappa_{\nu_{\text{ref}}}^{m,c}$ 'nin spektral indis değerini, m_{H_2} hidrojen molekülünün

kütlesini, $1/\zeta_{gas-dust}$ gaz ve toz oranını belirtir ve tüm bu parametreler kullanıcı tarafından belirlenir. Bu denklemde frekansın referans değeri 230 GHz ($v_{ref} = 230$ GHz), $1/\zeta_{gas-dust}$ oranı 1/100 olarak belirlenir. Bu denklem gaz ve tozun karışım halinde olduğu durumlarda geçerlidir.

$$\tau_d^{m,c}(v) = \tau_{d,ref}^{m,c} \left(\frac{v}{v_{ref}} \right)^{\beta^{m,c}} = (N_H^{m,c} \cdot \kappa_{v_{ref}}^{m,c} \cdot m_{H_2} \cdot \frac{1}{\zeta_{gas-dust}}) \cdot \left(\frac{v}{v_{ref}} \right)^{\beta^{m,c}} \quad (3.7)$$

Her molekül ve bileşenin tüm spektral çizgilerinin optik derinlik değeri Denklem 3.8 ile hesaplanmaktadır. Bu denklemde toplam sembolünde alt limit olarak kullanılan t ifadesi, verilen frekans aralığında molekülün tüm spektral çizgilerini belirtir. A_{ul} Einstein katsayısını; E_l , en düşük enerji seviyesini; g_u , bozunum üst limit değerini; $Q(m, T_{ex}^{m,c})$ ise molekülün bölüşüm fonksiyon değerini belirtir. Bölüşüm fonksiyonu, termodinamik dengede olan bir sistemin sıcaklık, hacim ve molekül sayısı gibi parametrelerinin tanımlandığı fonksiyondur. Moleküle ait tüm bilgiler laboratuvar ortamından elde edilerek veri tabanında kayıtlı olarak bulunur. Ayrıca, uyarım sıcaklık değeri olan $T_{ex}^{m,c}$ ve çizgi yoğunluk değeri olan $N_{tot}^{m,c}$ kullanıcı tarafından her molekül ve bileşen için ayrı bir şekilde molfit uzantılı dosya içerisinde tanımlanır.

$$\tau_l^{m,c}(v) = \sum_t \left[\frac{c_{light}^2}{8\pi v^2} A_{ul}^t N_{tot}^{m,c} \frac{g_u^t e^{-E_l^t/k_B T_{ex}^{m,c}}}{Q(m, T_{ex}^{m,c})} (1 - e^{-h\nu^t/k_B T_{ex}^{m,c}}) \phi^{m,c,t}(v) \right] \quad (3.8)$$

Denklem 3.8'deki gaz parçacıklarının termal hareketinden dolayı meydana gelen spektral çizgilerdeki genişleme ve mikro düzeyde gerçekleşen türbülansı hesaba katmak için t . spektral çizginin normalize edilmiş gauss profili Denklem 3.9 ile hesaplanır.

$$\phi^{m,c,t}(v) = \frac{1}{\sqrt{2\pi}\sigma^{m,c,t}} e^{-\frac{(v - (v^t + \delta v_{LSR}^{m,c,t}))^2}{2(\sigma^{m,c,t})^2}} \quad (3.9)$$

Her bir molekül ve bileşen için kaynak frekans değeri olan $\delta v_{LSR}^{m,c,r}$, kullanıcı tarafından MOLFIT uzantılı dosyada tanımlanan hız offset $\delta v_{offset}^{m,c}$ değeri ile ilişkili olduğu Denklem 3.10 ile gösterilmiştir.

$$\delta v_{LSR}^{m,c,t} = -\frac{\delta v_{offset}^{m,c}}{c_{light}} \cdot v^t \quad (3.10)$$

Bu denklemde v^t veritabanından elde edilen t . geçiş frekans değerini belirtir. Ayrıca, her bir molekül ve bileşen için standart sapma $\sigma^{m,c}$ değeri MOLFIT uzantılı dosyada hız $\Delta v_{width}^{m,c}$ genişlik değeri olarak Denklem 3.11’de belirtildiği gibi tanımlanmıştır.

$$\sigma^{m,c,t} = \frac{\frac{\Delta v_{width}^{m,c}}{c_{light}}(v^t + \delta v_{LSR}^{m,c,t})}{2\sqrt{2\ln 2}} \quad (3.11)$$

Sürekli ışın-ortalama arka plan sıcaklık değeri, kullanıcının her bir frekans aralığı için süreklilik katkısını ayrı ayrı tanımlayarak Denklem 3.12 ile parametrelendirilir. Bu denklemde v_{min} verilen frekans aralığında en düşük seviyeli frekans değerini belirtir. Kullanıcı tarafından tanımlanan T_{bg} ve T_{slope} , sırasıyla arka plan süreklilik sıcaklığını ve sıcaklık eğimini tanımlar. Tozun katkısının bulunması her zaman olası değildir. Bu nedenle, gelecekte bu durum bir seçenek olarak sunulur. Şu andaki mevcut durumun kapsamı dışında olduğunu varsayarak gelecekte oluşacak durumlar için kaynağın yapısını tanımlaması gerekir.

$$I_{bg}^{core}(v) = T_{bg} \left(\frac{v}{v_{min}} \right)^{t_{slope}} + J_{CMB} \quad (3.12)$$

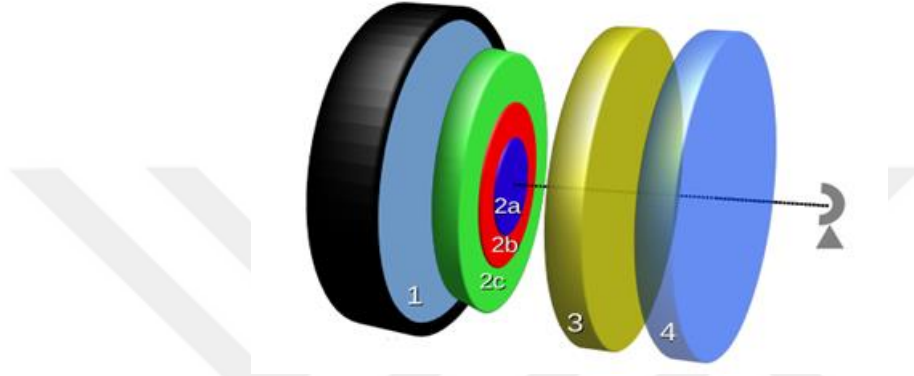
Denklem 3.1’deki süreklilik katsayı değeri, her bileşen için etkili bir toz sıcaklık değeri ile Denklem 3.5’deki kaynak fonksiyon kullanılarak Denklem 3.13 ile hesaplanır. Burada $T_{d,off}^{m,c}$ ve $T_{d,slope}^{m,c}$ değerleri molfit uzantılı dosyada kullanıcı tarafından belirlenir. Eğer bu değerler MOLFIT uzantılı dosyada tanımlanmamışsa tüm bileşenler için $T_d^{m,c} = T_{ex}^{m,c}$ olduğu varsayılır. Her bir bileşen için arka plan yoğunluk değerini ($\gamma \equiv 1$) tanımlamak için kullanıcı toz geçirgenlik değerini Denklem 3.7’yi kullanarak ve toz sıcaklık değerlerini Denklem 3.13 kullanarak tanımlamalıdır.

$$T_d^{m,c}(v) = T_{ex}^{m,c}(v) + \Delta T_d^{m,c}(v) = T_{ex}^{m,c}(v) + T_{d,off}^{m,c} \left(\frac{v}{v_{min}} \right)^{T_{d,slope}^{m,c}} \quad (3.13)$$

Son olarak Denklem 3.1’deki son terim olan J_{CMB} değeri tek cihaz ile yapılan gözlemler için kozmik arka planda bir yoğunluk değerini belirtir. İnterferometrik gözlemler için kozmik arka planın katkısı filtrelendir ve hata çıkarılması gerekir.

Çekirdek katmanların tersine ön plan bileşenleri Şekil 3.2’de gösterildiği gibi birbirleri ile etkileşim halindedir. Şekil 3.2’de, Denklem 3.12 ile hesaplanan ve $I_{bg}^{core}(v)$ yoğunluğuna sahip arka plan katmanlarının birincisinin önünde görüş hattına dik bir düzlemde 2a, 2b, 2c adında üç

farklı çekirdek bileşen bulunmaktadır. 3 ve 4 numaralı ön plan katmanları, görüş hattı boyunca (siyah kesikli çizgi ile belirtilen) çekirdek katmanlar ile teleskop arasında yer alır. Her bileşen, farklı renklerle gösterilen farklı uyarma sıcaklıkları, hız sapma değerleri ile tanımlanır. Her katmanın kalınlığı dolaylı olarak toplam kolon yoğunluk değeri $N_{tot}^{m,c}$ ile tanımlanır. Ön plan katmanlarının her biri için, ışın doldurma faktörünün 1 olduğunu varsayılır. Bu varsayım her ön plan katmanının tüm ışını kapladığı anlamına gelir.



Şekil 3.2: Siyah halka ile belirtilen Gauss şeklindeki bir ışının içindeki çekirdek ve ön plan katmanlarının dağılımı. (Möller ve Schilke, 2017)

Burada soğurma olayı, yalnızca emici katman için uyarım sıcaklığı arka planın sıcaklığından daha düşükse gerçekleşir. Ön plan bileşenleri yenilenebilir bir şekilde düşünülmelidir. Bu nedenle, ön plan katmanları için ışımsal transfer denkleminin çözümü için Denklem 3.1 kullanılmaz. Ön plan bileşenleri için ışımsal transfer denkleminin çözümü Denklem 3.14 ile verilir.

$$T_{mb}^{fore}(v)_{m,c=i} = \eta(\vartheta^{m,c=1})(S^{m,c=1}(v) - T_{mb}^{core}(v))(1 - e^{-\tau_{total}^{m,c=1}(v)}) + T_{mb}^{core}(v)$$

$$T_{mb}^{fore}(v)_{m,c=i} = \eta(\vartheta^{m,c=1})(S^{m,c=1}(v) - T_{mb}^{fore}(v)_{m,c=(i-1)})(1 - e^{-\tau_{total}^{m,c=1}(v)}) +$$

$$T_{mb}^{fore}(v)_{m,c=(i-1)} \quad (3.14)$$

Denklem 3.14'te m mevcut molekülün indisini; i , tüm ön plandaki bileşenler üzerinde çalışan bir indisi temsil eder. Ayrıca, her ön plan bileşeninin tüm ışını kapladığı ($\eta(\vartheta^{m,c=1} \equiv 1)$) varsayılır. Böylece Denklem 3.14 ve 3.15 şeklinde basitleştirilebilir. Denklem 3.15'te $T_{mb}^{core}(v)$ değeri, ışın-ortalama arka plan sıcaklık değeri $I_{bg}^{core}(v)$ ile birlikte çekirdek spektral veriyi belirtir. Ön plandaki diğer bileşenlerin katkısını hesaplamak için, öncelikle çekirdek nesnelerin katkısı hesaplanır. Toplam sütun yoğunluk değeri $N_{tot}^{m,c}$ bazı moleküllerin bulunma miktarına

ve o molekülü içeren katmanın kalınlığına bağlıdır. Görüş hattı boyunca bileşenlerin sırası, MOLFIT uzantılı dosyada bulunan bazı ön plan bileşenleri ile belirlenir.

$$T_{mb}^{fore}(v)_{m,c=1} = [S^{m,c=1}(v)(1 - e^{-\tau_{total}^{m,c=1}(v)}) + T_{mb}^{core}(v)e^{-\tau_{total}^{m,c=1}(v)}]$$

$$T_{mb}^{fore}(v)_{m,c=i} = [S^{m,c=i}(v)(1 - e^{-\tau_{total}^{m,c=i}(v)}) + T_{mb}^{core}(v)_{m,c=(i-1)}e^{-\tau_{total}^{m,c=i}(v)}] \quad (3.15)$$

İç içe geçmiş çizgiler ve optik derinlik değeri tüm molekülleri ve bileşenleri tek seferde eğri uydurma işleminden geçirerek elde edilir. Modelleme işlemi, XCLASS içerisinde bulunan iso_ratio adlı dosyadan elde edilen bir molekülün izotopları (daha yüksek titreşim durumları) ile eş zamanlı olarak yapılır.

Modelleme işleminin gerçekleştiği XCLASS'ın myXCLASS bölümü, modellenen spektrumu gözlemsel veriler ile karşılaştırırken enstrümantal çözünürlük efektlerini doğru bir şekilde hesaba katmak için, modellenen spektrumu her kanal üzerine ayrı ayrı entegre eder. Bu nedenle, myXCLASS, verilen her frekans değeri (v) sırasıyla her kanalın merkezini tanımlamış olur. Bu özellik Denklem 3.16 ile çözülür.

$$T_{mb}(v) = \frac{1}{\Delta v_c} \int_{v-\frac{\Delta v_c}{2}}^{v+\frac{\Delta v_c}{2}} T_{mb}(\tilde{\nu}) d\tilde{\nu} \quad (3.16)$$

Bu denklemde Δv_c , kanal genişlik değerini belirtir. Fakat Denklem 3.1 ve 3.15'in karmaşıklığından dolayı Denklem 3.16'daki entegrasyon yapılamaz. Bu nedenle, myXCLASS, Denklem 3.16 ile elde edilen değerleri toplayarak her bileşenin ve kanalın parçalı olarak entegrasyonunu gerçekleştirir. (Möller ve Schilke, 2017)

3.1.1. MOLFIT Dosyası

MOLFIT uzantılı dosyada, modelleme işleminin gerçekleştiği sırada kullanıcı tarafından hangi moleküllerin işleme alınacağını ve bu moleküllerin bileşen sayısı bulunur. Aynı zamanda bu dosya içerisinde belirlenen moleküllerin bileşenlerine ait, kaynak boyutu ($\theta^{m,c}$ - arcsec), uyarım sıcaklık değeri (T_{ex} - Kelvin), sütun yoğunluk değeri (N_{tot} - cm^{-2}), FWHM değeri (Δv - km s^{-1}), hız ofset değeri (V_{LSR} - km s^{-1}) ve bileşenin çekirdek mi ön plan mı olduğunu belirlemek amacı ile kullanılacak olan bayrak değeri (CFFlag - c ya da f) belirtilir. Şekil 3.3'te CS ve HCS adlı moleküller hakkında yukarıdaki bilgileri içeren örnek MOLFIT dosyası gösterilmektedir.

% Number of molecules = 2						
% size:	T_ex:	N_tot:	V_width:	V_off:	CFFlag:	
CS;v=0;	3					
48.470	300.00	3.91E+17	2.86	-20.564	c	
21.804	320.00	6.96E+17	8.07	30.687	c	
81.700	208.00	1.46E+17	5.16	-10.124	c	
HCS+;v=0; 2						
% size:	T_ex:	N_tot:	V_width:	V_off:	CFFlag:	
	150.00	1.10E+18	5.00	-0.154	f	
	200.00	2.20E+17	3.10	-2.154	f	

Şekil 3.3: Örnek MOLFIT dosyası. (Möller ve Schilke, 2017)

MOLFIT dosyası içerisinde herhangi bir molekül için parametrelerin tanımı, veri tabanında bulunan molekülün adıyla aynı olması gereken molekülün adını açıklayan bir satırla başlar, ardından bu molekül için bileşenlerin sayısı kadar satır ile devam eder. Hemen aşağısında bulunan her bileşen için ayrı ayrı oluşturulan satırlarda bileşenlerin parametreleri açıklanmaktadır. Tüm parametreler boşluklarla ayrılır, yorumlar ise % karakteri ile işaretlenir.

Uyarım sıcaklık değerinden (T_{ex}) farklı olarak her bileşenin toz sıcaklık değerini tanımlamak için MOLFIT dosyasına V_{off} ve CFFlag adlı sütunların arasına Denklem 3.13 kullanılarak hesaplanan toz geçirgenlik $T_{d,off}^{m,c}(T_{doff})$ ve toz sıcaklık değerini $T_{d,slope}^{m,c}(T_{dslope})$ belirten iki adet daha sütun eklenir. Bu durum için örnek MOLFIT dosyası Şekil 3.4'te gösterilmiştir. (Möller ve Schilke, 2017)

% Number of molecules = 2							
CS;v=0; 3							
% s_size:	T_rot:	N_tot:	V_width:	V_off:	T_doff:	T_dslope:	CFFlag:
48.47039	50.000	3.9E+16	2.862117	-2.5643	3.0	0.0	c
40.10603	56.531	2.3E+18	4.210278	-7.3161	3.0	0.0	c
29.09758	68.441	2.0E+17	4.024519	0.2130	2.5	1.0	c
	6.0857	2.2E+17	1.000000	-1.9000	3.0	0.0	f
HCS+;v=0; 1							
% s_size:	T_rot:	N_tot:	V_width:	V_off:	T_doff:	T_dslope:	CFFlag:
32.43532	43.234	2.1E+16	1.354217	-9.5211	4.0	0.1	c

Şekil 3.4: Toz geçirgenlik ve toz sıcaklık değerleri eklenmiş örnek MOLFIT dosyası. (Möller ve Schilke, 2017)

3.1.2. Iso Ratio Dosyası

İzotop, nötron sayısı ve kütle numarası farklılık gösteren bir atomun veya molekülün varyantına verilen isimdir. Iso ratio adlı dosya moleküller ve o molekülün izotopları arasındaki oranı gösterir ASCII formatında bir dosyadır. Bu dosya üç adet sütundan oluşur. İlk iki sütun molekülleri üçüncü sütun iki molekül arasındaki oranı tanımlar. Dosya içerisinde yorumlar "%" karakteriyle belirlenir, yani "%" ifadesinin sağ tarafındaki tüm karakterler dikkate alınmaz. (Möller ve Schilke, 2017)

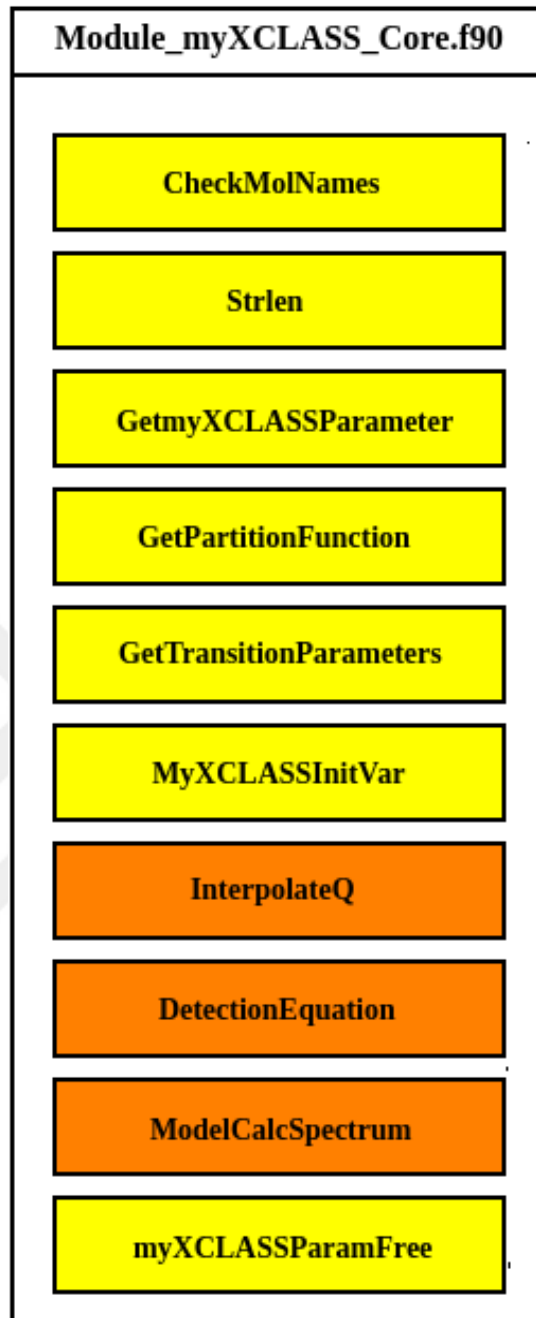
3.1.3. XCLASS'ın Hızlandırılması İşlemi

XCLASS içerisinde bulunan myXCLASS programı, bir boyutta izotermal bir nesne için sonlu kaynak boyutunu ve toz zayıflamasını göz önünde bulundurup ışımsal transfer denklemini çözerek verileri modellemektedir. Aynı zamanda verilen spektrumda çizgi-yoğun yerleri tanımlamak için tasarlanmıştır. myXCLASS programı, bir spektrumu keyfi sayıda molekül ile modelleyebilir ve böylece her bir molekülün nerede katkısı olduğunu tanımlayabilir. myXCLASS, aşağıda gösterilen 4 adet Fortran dosyasını içermektedir:

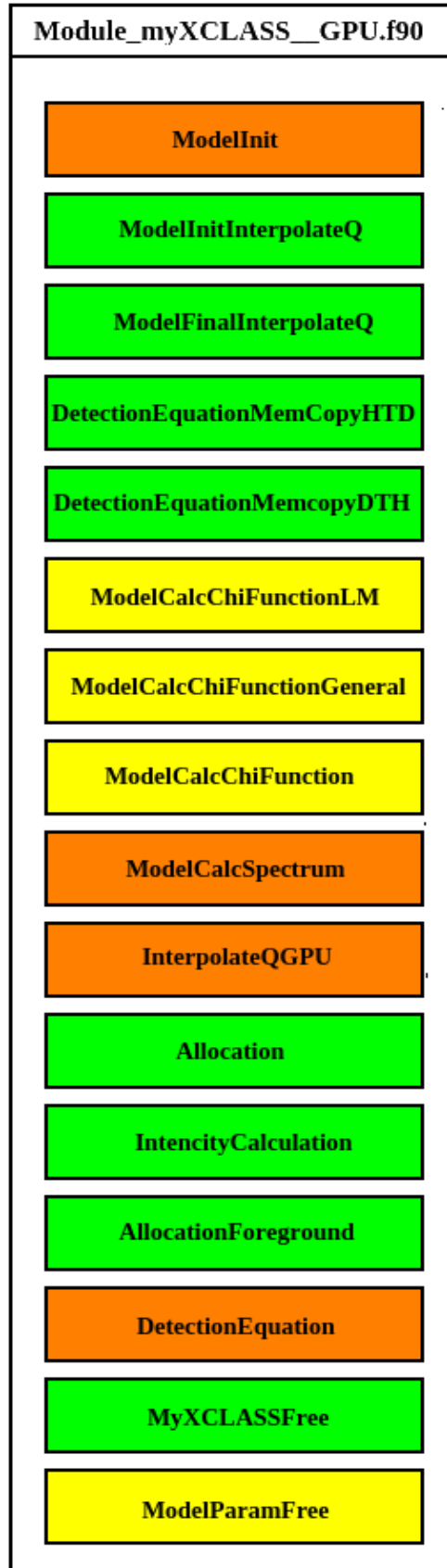
- Module_myXCLASS__Core.f90
- Module_myXCLASS__GPU.f90
- Module_myXCLASS__MPI.f90
- Module_myXCLASS__SMP.f90

Bu tez çalışmasının birinci kısmında, gerçekleştirilen modifikasyon işlemleri Module_myXCLASS_Core.f90 ve Module_myXCLASS_GPU.f90 dosyaları üzerinde yapılmıştır. Şekil 3.5 ve 3.6 sırası ile Module_myXCLASS_Core.f90 ve Module_myXCLASS_GPU.f90 dosyalarında bulunan fonksiyonları göstermektedir.

Şekil 3.5'deki Module_myXCLASS_Core.f90 dosyası içerisinde bulunan turuncu renkli InterpolateQ, DetectionEquation ve ModelCalcSpectrum adlı fonksiyonlar Module_myXCLASS_GPU.f90 adlı dosyaya taşınmıştır. Şekil 3.6'daki sarı renkli fonksiyonlarda herhangi bir değişiklik yapılmamıştır. Yeşil renkli fonksiyonlar yeni oluşturulmuştur. Turuncu renkli fonksiyonlar ise Module_myXCLASS_Core.f90 adlı dosyadan alınmış ve GPU programlama tekniği kullanılarak “çekirdek”e çevrilerek hızlandırılmıştır. Bu şekildeki ModelInitInterpolateQ, ModelFinalizeInterpolateQ adlı fonksiyonlar InterpolateQGPU adlı çekirdekler için yazılmıştır ve InterpolateQGPU'de kullanılacak olan cihaz (device) değişkenler için CPU'dan GPU'ya ve GPU'dan CPU'ya bellek kopyalama işlemlerinin yapıldığı fonksiyonlardır.



Şekil 3.5: Module_myXCLASS_Core.f90 dosyasında bulunan fonksiyonlar.



Şekil 3.6: Module_myXCLASS_GPU.f90 dosyasında bulunan fonksiyonlar.

DetectionEquationMemCopyHTD, DetectionEquationMemCopyDTH adlı fonksiyonlar ise DetectionEquation adlı fonksiyon tarafından çağrılan Allocation, IntensityCalculation ve AllocationForeground adlı çekirdeklerde kullanılacak olan cihaz değişkenleri için CPU'dan GPU'ya ve GPU'dan CPU'ya bellek kopyalama işlemlerinin yapıldığı fonksiyonlardır.

ModelCalcSpectrum adlı fonksiyon, verilen parametre vektörü için spektrum hesaplamaktadır. ModelCalcSpectrum içerisinde çağrılan InterpolateQ ve DetectionEquation fonksiyonlar spektrum hesabını hızlandırmak amacı ile CUDA FORTRAN programlama dili kullanılarak çekirdeğe dönüştürülmüştür. InterpolateQ fonksiyonunun ismi InterpolateQGPU olarak değiştirilmiştir. DetectionEquation adlı fonksiyon içerisinde birbirine bağlı çok fazla değişken bulunmaktadır. Bu nedenle fonksiyonun tamamını çekirdeğe dönüştürmek yerine hesap yoğun kısımlarını alınıp Allocation, IntensityCalculation, AllocationForeground adında yeni çekirdekler oluşturulmuştur.

Son olarak MyXCLASSFree adlı fonksiyon ise bütün Module_myXCLASS_GPU.f90 adlı dosya içerisinde bulunan cihaz dizilerinin program sonlandıktan sonra bellekte kapladığı alanı boşaltmak için oluşturulmuştur. ModelCalcSpectrum fonksiyon tarafından çağrılmaktadır. XCLASS içerisinde yapılan tüm bu değişikliklerin getirdiği performans kazancına ait veriler ve değerlendirmeler Bulgular bölümünde ayrıntılı olarak sunulmuştur.

3.1.3.1. ModelCalcSpectrum Fonksiyonu

Bu fonksiyon belirli bir parametre vektörü için spektrum hesaplamaktadır. ModelCalcChiFunction adlı fonksiyon içerisinde bütün parametre vektörleri için ayrı ayrı çağrılarak bir model spektrum ve gerçek spektrum ile model spektrum arasındaki hata değerini gösteren ch^2 vektörünü oluşturmaktadır. ch^2 vektörü Denklem 3.17 ile hesaplanmaktadır.

$$ch^2 = \sum_{i=0}^N \left[(y_i^{obs} - y_i^{fit})^2 \frac{1}{(\sigma_i^{error})^2} \right] \quad (3.17)$$

Bu denklemde, y_i^{obs} değeri i . noktadaki deneysel veriyi, y_i^{fit} ise i . deneysel veriye karşılık gelen değeri, σ_i^{error} de i . veri noktasının hatasını temsil etmektedir. (Sánchez-Monge, 2018) ModelCalcSpectrum adlı fonksiyonun giriş argümanları Şekil 3.7'de gösterilmiştir.

```

!< parameter vector
ParameterVector
!< flag for indicating if model function values are stored or not
ModelFunctionFlag
!< total number of componentsDetect
NumComp
!< total number of frequency ranges
TotalNumFreqRan
!< total number of transition frequencies
TotalNumTransFreq
!< chi2 value for parameter vector
chi2Value
!< total number of molecules
TotalNumMol
!< total number of data points
NumDataPoints
!< output variable: model function values
ModelFuncList
!< number of free parameters
NumParamVector

```

Şekil 3.7: ModelCalSpectrum giriş değerleri.

Eğri uydurma işlemi için seçilen molekül hakkındaki bilgilerin tutulduğu Molfit isimli bir dosya bulunmaktadır. Bu dosyada hangi molekül işlenecekse o molekül için bileşen sayısı tanımlanır. Kullanıcı bu dosya içerisinde molekülün her bir bileşeni için kaynak boyutu ($\theta^{(m,c)}$ - *arcsec*), uyarım sıcaklık değeri (T_{ex} - *Kelvin*), sütun yoğunluğu (N_{tot} - cm^{-2}), hız genişlik değeri (FWHM – Maksimum genlik değerinin yarısının genişliği) (Δv – $km s^{-1}$), hız denge değeri (V_{LRS} – $km s^{-1}$) ve bayrak değişken (CFFlag: bileşenin ön plan mı- *f* çekirdek mi- *c* olduğunu belirtir) değerlerini tanımlamak zorundadır. Bu dosya içerisinde bulunan değerler myXCLASSParameter adlı global dizide tutulmaktadır.

Iso ratio dosyasında ise moleküller ve izotopları arasındaki izotop oranları tanımlanmıştır. ASCII formatındaki dosyanın ilk iki sütunu sırasıyla molekülleri gösterir. Üçüncü sütun, her iki molekül arasındaki izotop oranını gösterir. Bu dosya içerisinde bulunan değerler IsoRatioConversionTable adlı dizide tutulmaktadır. ModelCalcSpectrum fonksiyonunun işlem adımları şu şekildedir:

- 1- Molfit dosyasının parametreleri güncellenir.
- 2- Bütün rotasyon sıcaklıkları ve doppler-kayması geçiş frekansları için bölüşüm fonksiyon değeri belirlenir.
- 3- Uyarım sıcaklık değeri kontrol edilir.
- 4- InterpolateQGPU çekirdeği çağrılır.

5- Çıkış değişkenleri için bellekte alan tahsis edilir.

6- Emisyon bileşenlerinin sayısı belirlenir

7- Işın boyutu belirlenir:

Tek çanaklı veri için (InterFlag(l) = .false.) ışın boyutu aşağıdaki Denklem 3.18 ile belirlenir.

$$\theta_t = 1,22 * c/vD * \zeta = 1,22 * c/vD * (180. d0 * 3600. d 0/pi) \quad (3.18)$$

Teleskop ışınının FWHM boyutu,

D , teleskopun çapını,

c , ışık hızını belirtir.

İnterferometrik veri için (InterFlag(l) = .true.) ışın boyutu Denklem 3.19 ile belirlenir.

$$\theta_t = D \quad (3.19)$$

D , kullanıcı tarafından belirlenen interferometrik ışının FWHM boyutudur ve tüm frekans aralığında sabittir.

8- Maksimum ışın dolgu faktörünü ve tüm emisyon bileşenlerinin ilgili bileşen indeksini belirlenir.

9- Ön plan ve arka plandaki bileşenlerin yoğunluk değeri hesaplanır.

10- DetectionEquation fonksiyonu çağrılır.

11- Bütün bileşenlerin toplam yoğunluğu hesaplanır.

12- Modellenmiş fonksiyon çıkış dosyasına yazılır.

13- Hata'yı belirten χ^2 değeri hesaplanır.

3.1.3.2. InterpolateQGPU Fonksiyonu

İnterpolasyon, bir dizi veri noktası arasına yeni veri noktaları oluşturma yöntemidir. Astronomide gözlem yoluyla elde edilen çok sayıda veri noktası olduğundan bu bağımsız veri

noktaları arasında bağlantı kurmak için interpolasyon işleminden geçirilmesi gerekmektedir. (Hall ve Meyer, 1976) InterpolateQGPU, doğrusal interpolasyon işleminin gerçekleştirildiği çekirdektir. Verilen sıcaklık aralığına göre bölüşüm fonksiyonu interpolate edilmektedir. İlk önce uyarım sıcaklık değerinin logaritması Denklem 3.20 ile hesaplanmaktadır.

$$x = \log_{10}(\text{temperature}) \quad (3.20)$$

Elde edilen değer verilen en düşük sıcaklık değerinden daha az ise en düşük sıcaklık değeri bölüşüm fonksiyon değerine atanmaktadır. Verilen en yüksek sıcaklık değerinden daha yüksek ise en yüksek sıcaklık değeri bölüşüm fonksiyon değerine atanmaktadır. Daha sonra doğrusal interpolasyon işlemi başlamaktadır. Doğrusal interpolasyon hesaplamak için Denklem 3.21’de gösterilen formül kullanılmıştır. Denklemde $f(x)$, bölüşüm fonksiyon değerini temsil etmektedir ve 2 boyutlu cihaz değişkeni olarak tanımlanmıştır.

$$f(x) = f_0 + \frac{(f_1 - f_0)}{(x_1 - x_0)} * (x - x_0) \quad (3.21)$$

Bu denklemde $f(x) = [\text{toplam bileşen sayısı}, \text{toplam molekül sayısı}]$ ’dır. Ayrıca, bölüşüm fonksiyon değişkeni için sıcaklık, $\lg Q$ ve bölüşüm fonksiyon değerlerini gösteren diziler cihaz olarak sırasıyla Şekil 3.8’deki gibi tanımlanmıştır. Diğer değişkenler ise çekirdek içerisinde tanımlanmış yerel değişkenlerdir.

```
!< temperatures for partition function
TempPartFunc_d = [NumberOfTemperatures]
!< lgQ entries of table PartitionFunction
lgQ_d = [NumberOfTemperatures, NumberMoleculePartFunc]
!< partition function at the rotation temperatures
Olocal = [NumComp, TotalNumMol]
```

Şekil 3.8: InterpolateQGPU fonksiyonunun cihaz dizileri.

Interpolate adlı fonksiyon tamamen CUDA Fortran programlama dili kullanılarak çekirdeğe dönüştürülmüştür. GPU için yapılan ayarlamalar aşağıdaki gibidir:

```
tBlock = dim3(MaxNumThreads, 1, 1)
```

```
grid = dim3(ceiling(real(NumberOfTemperatures)/tBlock%x), 1, 1)
```

MaxNumThreads sayısı, bir önceki bölümde belirtildiği gibi Tesla K20m GPU kartında 1024 tür. Yapılan ayarlamalar cihaz olarak tanımlanan dizilerin boyutları ve Tesla K20m GPU

kartının özellikleri göz önünde bulundurularak yapılmıştır. Çekirdeğin tamamı GPU üzerinde çalışmaktadır. Elde edilen performans kazanımı, Bulgular bölümünde ayrıntılı olarak sunulmuştur.

3.1.3.3. DetectionEquation Fonksiyonu

Nesnenin fiziki yapısını öğrenmek için nesneden gelen ışığın spektrumunu incelemek en önemli yöntemlerden biridir. Nesnenin sıcaklık, kimyasal kompozisyonu, hareketi, kökeni ve evrimi ve diğer ipuçları gibi pek çok parametre spektrum analizi ile belirlenir. myXCLASS adlı program DetectionEquation fonksiyonu ile bir boyutlu izotermal bir nesne için ışımsal transfer denklemini çözerek spektrum modellemektedir. Fonksiyon, belirli bir frekans aralığında ve bir boyutta izotermal nesne için ışımsal transfer denklemini çözerek bir spektrum modeli oluşturur. Eğer transfer işlemi verilen frekans aralığında bulunmuyor ise fonksiyon bir şey yapmadan sonlandırılır. DetectionEquation adlı fonksiyonda, nesneden gelen ışığın spektrumdaki tek bir Gauss eğrisi Denklem 3.22 ile hesaplanmaktadır. m değeri molekülü, c değeri ise molekül içinde bulunan bileşenleri, $\eta(\vartheta)^{m,c}$ molekül ve bileşenin ışın doldurma faktörünü, $(\vartheta)^{m,c}, \vartheta_t$ değerleri ise sırası ile kaynak ve teleskop ışın FWHM boyutunu temsil etmektedir.

$$(\eta(\vartheta)^{m,c}) = \frac{(\vartheta^{m,c})^2}{\vartheta_t(\nu)^2 + (\vartheta^{m,c})^2} \quad (3.22)$$

İkinci adım, Rayleigh-Jeans sıcaklık değerinin belirlenmesidir. Fizikte Rayleigh-Jeans, klasik argümanlar aracılığı ile belirli bir sıcaklıkta siyah bir gövdeden dalga boyunun bir fonksiyonu olarak elektromanyetik radyasyonun spektral radyasyonuna bir yaklaşımdır. (George ve Lightman, 1985) Üçüncü adımda, toz katkı değeri aşağıdaki Denklem 3.23 ile hesaplanır:

$$\tau_d^{m,c} = N_H^{m,c} * \kappa_{\nu_{ref}}^{m,c} * m_{H_2} * \left(\frac{1}{\zeta_{gas-dust}} \right) * \left(\frac{\nu}{\nu_{ref}} \right)^{(\beta^{m,c})} \quad (3.23)$$

Bu denklemde, N_H hidrojen yoğunluğunu, $\kappa_{\nu_{ref}}^{m,c}$ toz kütle bulanıklık değerini, β spektral indeksi, $\nu_{ref} = 230 \sim \text{GHz}$ bir dalga boyunun frekansını, m_{H_2} hidrojen molekülünün kütlesini gösterir.

$\frac{1}{\zeta_{gas-dust}}$ Değeri toz, gaz kütle oranını belirtir ve 1/100 değeri atanır. Dördüncü adımda her molekül m ve bileşen c için optik derinliği belirleme işlemi aşağıdaki denklem 3.24 ile yapılmaktadır:

$$\tau(\nu)^{m,c} = \sum \left[\frac{c_{light}^2}{8\pi\nu^2} * A_{ul}^t * N_{tot}^{m,c} * \left(\frac{g_u^t e^{-E_l^t/k_B T_{ex}^{m,c}}}{Q(m, T_{ex}^{m,c})} \right) * (1 - e^{-h\nu^t/k_B T_{ex}^{m,c}}) \phi(\nu)^{m,c,t} \right] \quad (3.24)$$

Burada, A_{ul} , Einstein katsayısını, u ve l indeksleri sırasıyla üst ve alt geçiş durumunu belirtir. E_l , en düşük enerji durum; g_u , üst sınır dejenerasyon; $Q(m, T_{ex}^{m,c})$, bölüşüm fonksiyon değerleridir. Bu değerler InterpolateQGPU adlı çekirdekten elde edilen değerlerdir. $T_{ex}^{m,c}$ ve N_{tot} değerleri kullanıcı tarafından tanımlanan ve molfit dosyasından alınan uyarım sıcaklık ve yoğunluk değeridir. Bir sonraki adımda denklem 3.25 ile kaynak fonksiyon hesaplanır, $S^{m,c}(\nu)$ ifadesi, kaynak fonksiyonunu tanımlar ve Kirchhoff'un termal radyasyon yasasına göre yapılır.

$$S^{m,c}(\nu) = \frac{\tau_l^{m,c}(\nu) J(T_{ex}^{m,c}, \nu) + \tau_d^{m,c}(\nu) J(T_d^{m,c}, \nu)}{\tau_l^{m,c}(\nu) + \tau_d^{m,c}(\nu)} \quad (3.25)$$

Şekil 3.9'da belirtilen diziler cihaz olarak tanımlanmıştır. GPU için yapılan ayarlamalar aşağıdaki gibidir.

tBlock = dim3(MaxNumThreads, 1, 1)

grid = dim3(ceiling(real(IsoCounter)/tBlock%x), 1, 1)

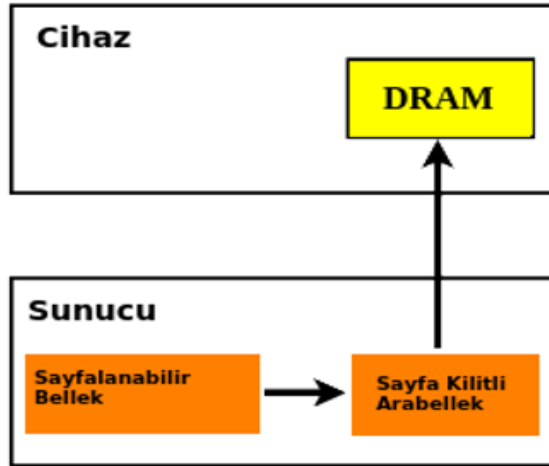
```
!< list containing all observational data
ObservationalDataList_d = [TotalNumberOfDataPoints, 3]
!< working array for integrated line intensity
IntensityPerPeak_d = [TotalNumberOfFrequencyRanges, j, MaxNumIso, k]
!< working array for integrated line intensity
PrevIntPerPeak_d = [MaxNumIso, k]
!< working array for intensity per component calculation
LocalIntArray_d = [MaxNumIso]
!< working array for integrated line intensity
IntegHelperArray1_d = [MaxNumIso, k]
!< working array for integrated line intensity
IntegHelperArray2_d = [MaxNumIso, k]
!< working array for intensity per component calculation
TauHelperArray1_d = [MaxNumIso]
!< working array for intensity per component calculation
TauHelperArray2_d = [MaxNumIso]
```

Şekil 3.9: DetectionEquation fonksiyonunun cihaz dizileri.

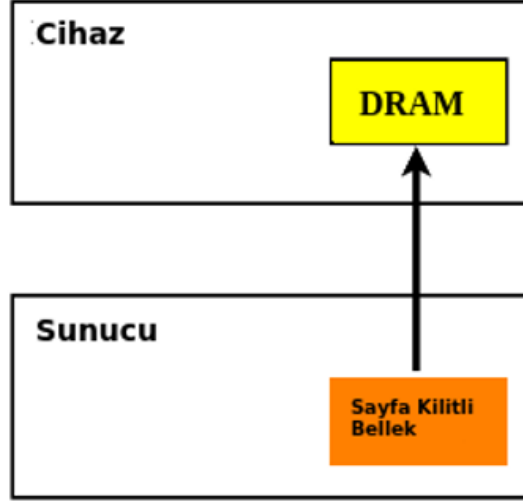
3.1.4. Optimizasyon İşlemi

GPU kartlar birçok bellek türüne sahiptir. Etkili bir programlama için bu bellek türlerinin verimli bir şekilde kullanılması gerekir. Sunucu ve cihaz belleği arasında veri aktarımı ve cihaz içinde farklı bellek bölümleri arasındaki veri aktarımı olmak üzere veri transfer işlemi iki bölümden oluşur. Çalışma süresi açısından herhangi bir hızlanma göstermeyen bazı işlemler

olabilir. Eğer sunucuda işlemin yürütülmesi, sunucu ve cihaz arasında çok fazla transfer gerektiriyorsa, işlemlerin sunucu üzerinde gerçekleşmesi daha avantajlı olur. Bu nedenle, en iyi uygulama performansı için, mümkün olduğunca sunucu cihaz arasındaki veri transferlerini en aza indirmek ve aynı zamanda bu tür transferler gerektiğinde optimize edildiğinden emin olmak önemlidir. Bu optimize işlemi sayfa kilitli (pinned) bellek kullanımını gerektirir, küçük aktarımları toplu olarak gerçekleştirmeyi ve veri aktarımlarını asenkron olarak gerçekleştirmeyi içermektedir. Normalde sunucuda değişkenler sayfalanabilir (pageable) bellekte bulunur. Sayfalanabilir bellek, programın sunucuda bulunan RAM'de mevcut olandan daha fazla alan kullanmasına izin vermek için programı diske götürülebilen bellektir. Veriler, sunucu ve cihaz arasında transfer edildiğinde, GPU üzerindeki Direct Memory Access (DMA) motoru, sayfa kilitli sunucu belleğini hedeflemelidir. Sayfalanabilir sunucu belleğinden GPU'ya veri aktarım işlemi sırasında, Şekil 3.10'da gösterildiği gibi sunucu işletim sistemi önce geçici bir sayfa kilitli arabellek tahsis eder, verileri sayfa kilitli arabelleğe kopyalar ve daha sonra diziler cihaza aktarır [17]. Sayfa kilitli arabellekler, cihazdan sunucuya veri transfer işlemini benzer şekilde gerçekleştirir. Eğer sunucudan cihaza transfer edilecek diziler sayfa kilitli olarak tanımlanırsa, Şekil 3.11'de gösterildiği gibi sayfalanabilir bellek ve sayfa kilitli arabellek arasındaki transferin maliyeti önlenmiş olur ve direkt olarak bellek transfer işlemi gerçekleşir. (Ruetsch ve Fatica, 2014)



Şekil 3.10: Sayfalanabilir bellekten sunucudan cihaza veri transferi. (Ruetsch ve Fatica, 2014)



Şekil 3.11: Sayfa kilitli bellekte sunucudan cihaza veri transferi. (Ruetsch ve Fatica, 2014)

CUDA FORTRAN’da sayfa kilitli bellek değişkeni pinned ve allocatable niteleyici kullanarak tanımlanmalıdır. allocatable niteleyicinin kullanılmasının sebebi sayfa kilitli bellekte alan tahsis işlemi gerçekleşip gerçekleşmediğini kontrol etmektir. Bellek tahsis işleminin başarısız olduğu durumda sayfalanabilir bellekte tahsis işlemi gerçekleşir. Bu çalışmada kullanılacak olan Tesla K20 GPU kartın veri transfer hızı Tablo 3.1’de gösterilmiştir.

Tablo 3.1: Tesla K20 veri transfer hızı. (Ruetsch ve Fatica, 2014)

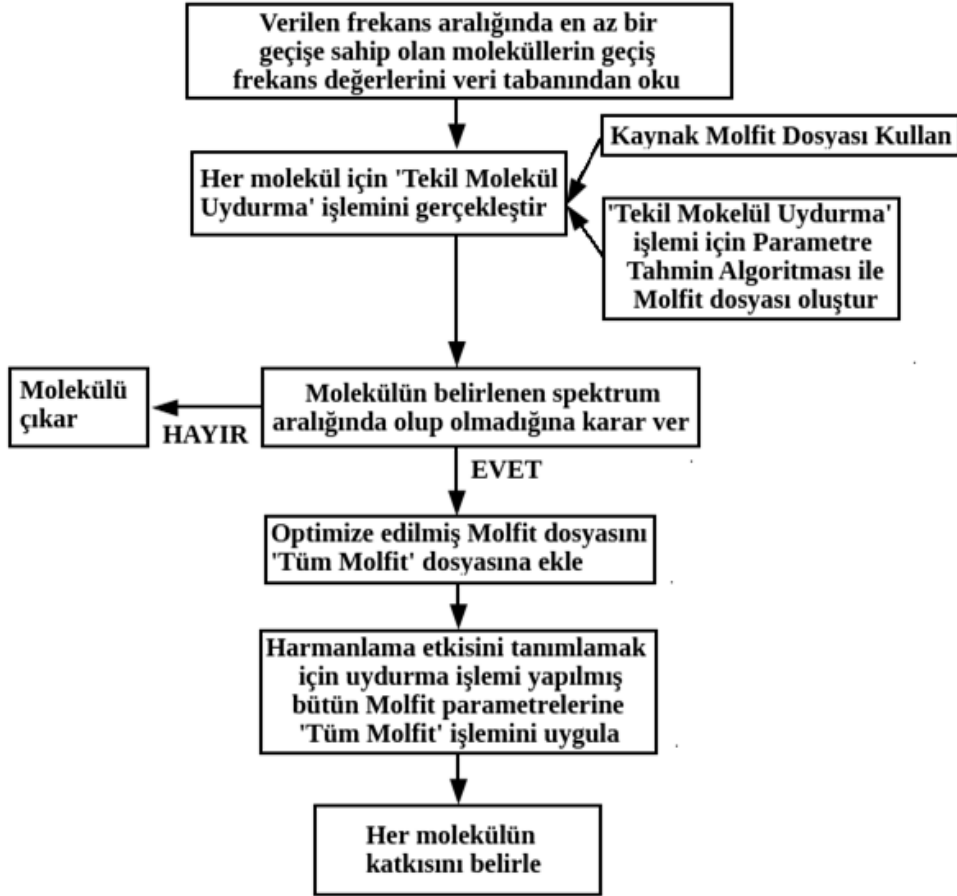
Sayfalanabilir Transfer		Sayfa Kilitli Transfer	
Sunucudan Cihaza bant genişliği (GB/s):	1,6595	Sunucudan Cihaza bant genişliği (GB/s):	5,7450
Cihazdan sunucuya bant genişliği (GB/s):	1,5933	Cihazdan sunucuya bant genişliği (GB/s):	6,5663
Transfer Boyutu (MB):		16.00000	

Bu çalışmada **InterpolateQGPU** adlı fonksiyonda bulunan *TempPartFunc* ve *lgQ*; **DetectionEquation** adlı fonksiyonda bulunan *TauHelperArray1*, *TauHelperArray2*, *LocalIntArray*, *IntegHelperArray1*, *IntegHelperArray2*, *PrevIntPerPeak*, *IntensityPerPeak* adlı diziler sayfa kilitli olarak tanımlanmıştır. Sayfa kilitli bellek kullanımı, ModelCalcSpectrum adlı fonksiyonunun ortalama çalışma süresini 1,44 saniyeden 0,96 saniyeye düşürmüştür.

3.2. SPEKTRAL ÇİZGİLERİN TESPİTİ

3.2.1. Otomatik Çizgi Tanıma Yazılımı

Otomatik çizgi tanıma yazılımı verilen bir spektrumda molekülleri otomatik olarak tanıma işlemini gerçekleştirir. Otomatik çizgi tanıma yazılımının algoritması iki aşamadan oluşur. Algoritma, kullanıcının belirlediği frekans aralığında en az bir geçişe sahip bütün molekülleri belirleyerek başlar. “SelecetMolecules” adlı bir giriş parametresi oluşturulur. Bu parametreye atanan moleküller ile hangi moleküllerin verilen frekans aralığında aranacağını ve diğer moleküllerin işleme alınmayacağına karar verilmiş olunur. Daha sonra Otomatik Çizgi Tanıma Yazılımı her bir molekülün spektrumunda ne kadar bulunduğunu hesaplamak için Tekli Molekül Uydurma (single molecule fit) işlemine başlar. Tekli Molekül Uydurma işlemi bittikten sonra Otomatik Çizgi Tanıma Yazılımı “Tüm Molekül Uydurma” (overall fit) işlemini gerçekleştirir. Otomatik çizgi tanıma yazılımının akış şeması Şekil 3.12’de gösterilmiştir.



Şekil 3.12: Otomatik Çizgi Tanıma yazılımı akış şeması.

Evrende çok sayıda molekül ve atom bulunmaktadır. Bu tez çalışmasında tüm moleküllerin içeriğini bulunduran 2.5 GB boyutundaki veri tabanında bulunan her molekülün veya atomun tek tek katkısını hesaplamak mümkün değildir. Bu nedenle kullanıcı tarafından tanımlanan frekans aralığında bulunma olasılığı yüksek olan moleküllerin listesini tutan StrongMolecules isminde bir liste tanımlanır. Daha sonra StrongMolecules adlı listede bulunan moleküller hakkında bilgi tutan bir optimized.molfit adında dosya oluşturulur. Otomatik Çizgi Tanıma Yazılımı ilk olarak bulunma olasılığı yüksek olan bu moleküller için “Tekli Molekül Uydurma” işlemini gerçekleştirir. Eğer molekül verilen frekans aralığında bulunuyor ise o molekülün optimized molfit dosyasında bulunan bilgileri strongMolecule.molfit isimli dosyaya aktarılır. StrongMolecules adlı listede bulunan tüm moleküller için bu işlem aynı şekilde ayrı ayrı gerçekleştirilir. Listedeki bulunan tüm moleküller için “Tekli Molekül Uydurma” işlemi bittiğinde oluşturulan strongMolecule.molfit dosyası all.molfit adlı dosyaya aktarılır. Bu dosya “Tüm Molekül Uydurma” işlemi sırasında tekrardan optimize edilir.

3.2.1.1. Tekli Molekül Uydurma İşlemi

Otomatik Çizgi Tanıma Yazılımı “Tekli Molekül Uydurma” işlemini Parametre Tahmin Algoritmasını kullanarak gerçekleştirir.

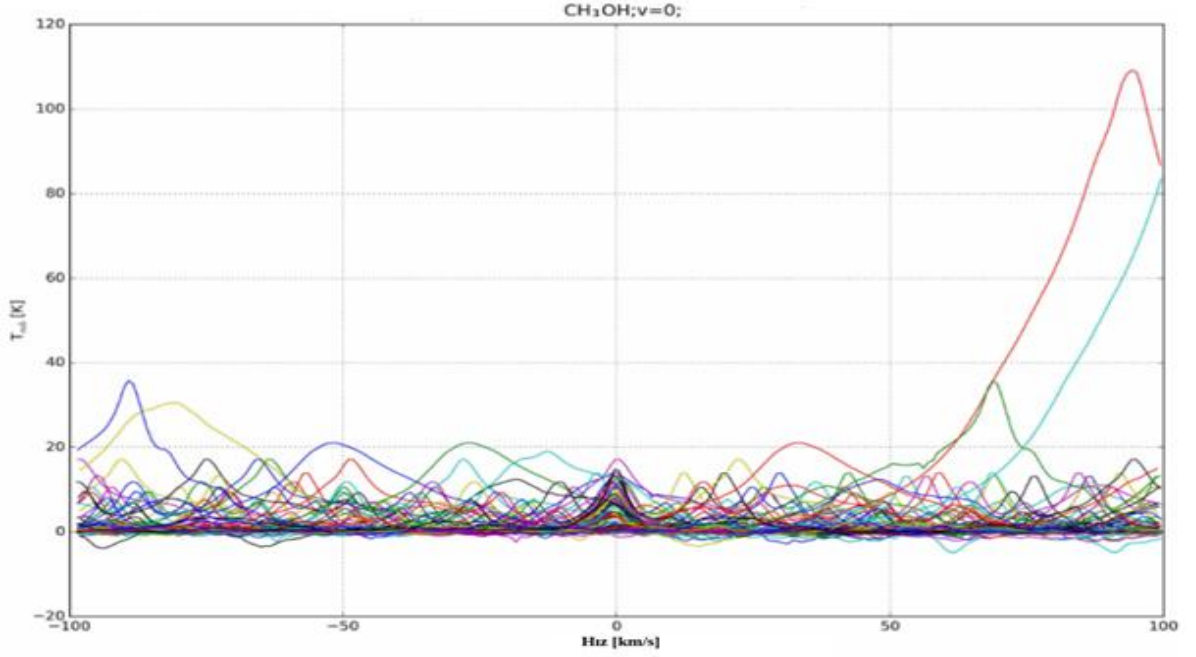
3.2.1.2. Parametre Tahmin Algoritması

Parametre Tahmin Algoritması gözlem yapılan bölgeden elde edilen spektrumun kullanıcı tanımlı frekans aralığında bulunan her molekülün bileşen sayısını ve bu bileşenlerin hız ofset değerlerini tespit eder. Elde edilen bu değerler StrongMolecule listesinde bulunan her bir molekül için uygulanacak olan “Tekli Molekül Uydurma” işleminde kullanılır. Şekil 3.13’te Parametre Tahmin Algoritmasının akış şeması gösterilmiştir.



Şekil 3.13: Parametre Tahmin Algoritması akış şeması.

Parametre Tahmin Algoritması her molekülün verilen frekans aralığında tüm geçiş frekanslarını veri tabanından okumakla başlar ve her geçiş frekansının etrafındaki gözlemsel verileri çıkarır. Bu işleme “geçiş frekansının spektrum penceresi” olarak adlandırılır. Şekil 3.14’te örnek olarak $v=0$ km/s ‘de CH_3OH adlı molekülün belirli bir frekans aralığında “geçiş frekansının spektrum penceresi” gösterilmiştir. Burada ayrı renklerde gösterilen her geçiş CH_3OH molekülüne aittir.



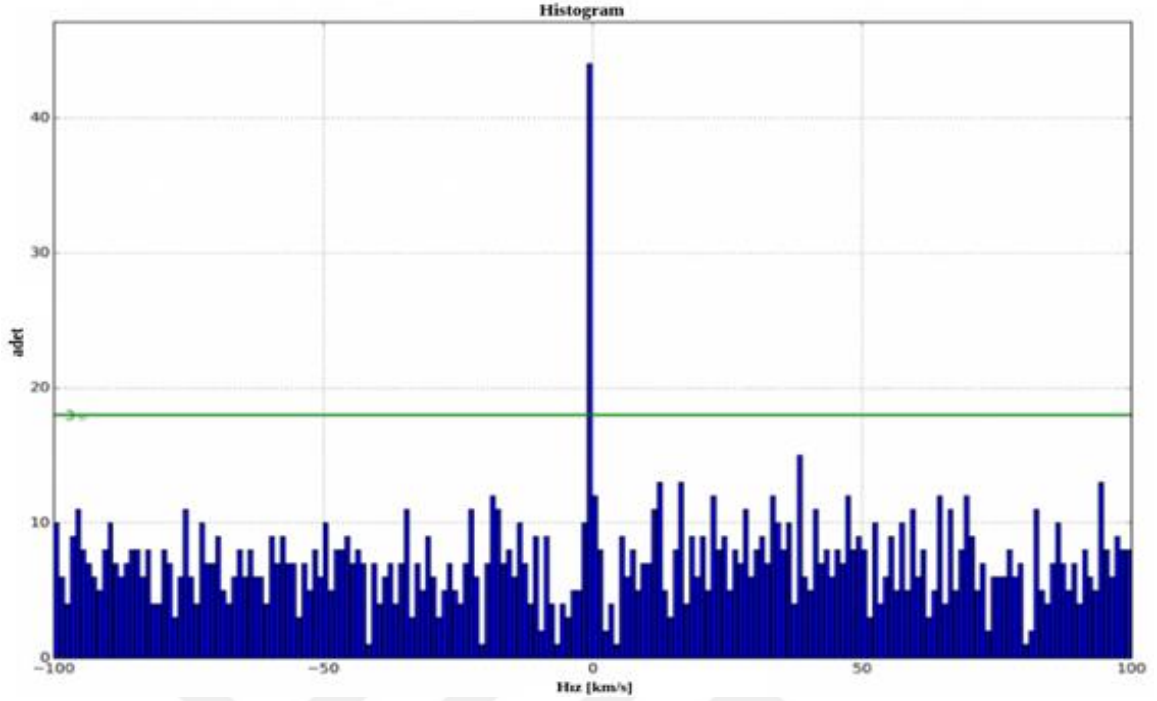
Şekil 3.14: CH₃OH adlı molekülün $v=0$ km/s'de geçiş frekans spektrum penceresi. (Möller, 2018)

Parametre tahmin algoritması Denklem 3.26 ile her geçişte bulunan sürekli spektrum çıkarılır. Bu denklemde T_{back} ve T_{slope} değerleri kullanıcı tarafından belirlenir. Doppler denklemi olarak da bilinen Denklem 3.27 ile frekansa bağlı sıcaklık grafiği, hız-sıcaklık grafiğine çevrilir.

$$I_{bg}^{core}(v) = T_{bg} \cdot \left(\frac{v}{v_{min}}\right)^{T_{slope}} + J_{CMB} \quad (3.26)$$

$$v = v_{LSR} + \frac{c}{v_{Rest}} (v_{Rest} - v) \quad (3.27)$$

Denklem 3.27'de c ışık hızını (km/s), v_{Rest} ise $v_{LSR}=0$ 'da molekülün geçiş frekans değerini göstermektedir. Daha sonra algoritma tüm geçiş frekansın oluşturduğu çizgilerin konumlarını ve yükseklik değerlerini belirler. Elde edilen yükseklik ve konum değerleri ile molekülün yoğunluğu ve hızı hakkında bilgi olunur. Algoritma geçiş frekanslarından kaç adet olduğunu ve konumlarını hesaplamak için histogram çıkarılır (Örnek Şekil 3.15). Algoritmada molekülün ve bileşenlerin yoğunluğunu tespit etmek için kullanıcı tarafından belirlenen frekans aralığı belirli aralıklara bölünür. Amaç hesap yoğunluğunu azalmaktır. Kullanıcı tarafından tanımlanan $VelBin$ adında parametre ve gürültü seviyesi değeri ile hız değerleri belli aralıklara bölünerek geçiş değerlerinin histogramı hesaplanır. Tüm bu işlemler sadece emisyon çizgileri için gerçekleştirilir.



Şekil 3.15: Şekil 3.14'te gösterilen CH3OH'nin geçiş frekans spektrum penceresinin merkezi hızlarının histogramı. (Möller, 2018)

Bileşenlerin sayısını ve karşılık gelen hız sapmalarını tanımlamak için 3σ değeri hesaplanır. 3σ sınırının üzerinde geçiş frekansa sahip tüm hızlar dikkate alınır ve sırasıyla emisyon çizgileri için hız bileşenlerinin sayısı hesaplanır.

Daha sonra Parametre Tahmin Algoritması, AlgorithmXMLFileEstParam parametresi ile her bileşen için diğer molfit parametreleri tahmin edilir ve böylece molekülün bileşeninin her parametresi için Doppler etkisi de göz önüne alınarak alt ve üst sınır değerler tanımlanır. Mevcut molekül ve bileşen için eğri uydurma işlemi gerçekleştirilirken seçilen alt ve üst sınır değerlerinde başka molekül ve bileşen olmadığı farz edilmiştir. Çünkü diğer moleküller olsa bile katkısı çok küçük olduğu için göz ardı edilebilir. Bu nedenle, parametre tahmin algoritması başlangıç değerlerini değil, sadece alt ve üst sınır değerlerini tanımlaması yeterlidir. Son olarak gözlemsel veri eğri uydurma işleminden geçirilerek, her bileşenin kaynak boyutu, uyarım sıcaklığı, sütun yoğunluğu ve hız genişliği değerleri hesaplanmış olur.

Algoritma, eğri uydurma işlemini her bileşen için ayrı ayrı uygular. Tüm emisyon bileşenleri için eğri uydurma işlemi bittikten sonra, algoritma, mevcut molekülün tüm emisyon bileşenlerinin tüm parametrelerinin eş zamanlı olarak optimize edildiği bir eğri uydurma işlemi AlgorithmXMLFileSMF parametresi tarafından kontrol edilerek gerçekleştirir. Emisyon

bileşenleri için gerçekleştirilen eğri uydurma işlemi daha sonra soğurma bileşenleri için gerçekleştirilecek eğri uydurma işleminde sürekli spektrum olarak kabul edilir. MinColumnDensityEmis ve MinColumnDensityAbs giriş parametrelerini kullanarak kullanıcı, emisyon ve soğurma bileşenleri için minimum kolon yoğunluklarını ayrı ayrı tanımlanır. MinColumnDensityEmis ve MinColumnDensityAbs giriş parametre değerlerinden yüksek yoğunluğa sahip bileşenler overall.molfit dosyasına eklenir. Böylece AlgorithmXMLFileSMF adlı giriş parametresi ve MOLFIT dosyası kullanılarak tekli molekül uydurma işlemi gerçekleştirilir.

Özet olarak, parametre tahmin algoritması, gözlemsel verilerden hız bileşenlerinin sayısını belirleyebilir ve hız bileşenlerine karşılık gelen hız ofset değerlerini tespit eder. Bu işlemi gerçekleştirirken molekülün veya bileşenin en tepe noktasını bulmakla kalmaz aynı zamanda iç içe geçmiş emisyon veya soğurma çizgilerinin yalnızca farklı hız sapmalarına sahip bileşenleri tespit edebilir. Ancak algoritma, benzer hız ofset değerlerine sahip ancak farklı uyarma sıcaklıklarına sahip bileşenleri tespit edememektedir. Algoritmanın belirli bir frekans aralığında çalışması ve bu frekans aralığının belirlenen aralıklara bölünerek tekli molekül uydurma işleminin gerçekleşmesi hesap-yoğunluğunun azalmasına neden olur.

3.2.1.3. Tespit Edilen Moleküllerin Kontrolü

Tekli Molekül uydurma işlemi bittikten sonra oluşturulan model spektrumda bütün frekans aralıklarında belirlenen gürültü seviyesinden yüksek en az bir çizgi olup olmadığı kontrol edilir. (Gürültü seviyesinden düşük seviyeye sahip olan tüm çizgiler göz ardı edilir.) Daha sonra model spektrumda gürültü seviyesinin yukarısında olan fakat gözlemsel veride bir karşılığı olmayan yapay çizgi olup olmadığı kontrol edilir. Bu nedenle kullanıcı tarafından MaxOverestimationHeight adında bir global tahmin eşik değeri yüzde cinsinden tanımlanır. MaxOverestimationHeight eşik değerinden daha büyük değere sahip çizgiler varsa, bir çizgi eklenir. Bu değer tüm moleküller için geçerlidir. Otomatik Çizgi Tanıma yazılımı model spektrum ile gözlemsel verinin yoğunluklarını Doppler etkisi (Denklemler 3.28) göz önünde tutarak karşılaştırır.

$$v_{\text{Doppler}}^i = v_t^i \cdot \left(1 - \frac{v_{\text{offset}}^{m,c}}{c_{\text{light}}}\right) \quad (3.28)$$

Denklem 3.28’de veri tabanından alınan i ’ninci geiş frekans deęerini, $v_{offset}^{m,c}$ molekölün molfit dosyasından alınan bileşenin hız offset deęerini belirtir.

$$\eta_{Peak}^i = \frac{I_{model}(v_{Doppler}^i)}{I_{observed}(v_{Doppler}^i)} \times 100 \quad (3.29)$$

Denklem 3.28’den elde edilen deęer ile aynı frekanstaki model spektrum ile gözlemsel verinin yoğunluęunun deęeri Denklem 3.29 ile hesaplayarak Otomatik izgi Tanıma yazılımı o molekölün belirlenen frekans aralıęında olup olmadıęına karar verir.

Elde edilen izgi sayısı, mevcut molekölün toplam geiş frekansı $v_{Doppler}$ sayısı karşılaştırıldıęında kullanıcı tanımlı tolerans deęerinden büyük ise moleköl verilen frekans aralıęında bulunmadı olarak kabul edilir.

Model spektrum sayısı, gözlemlenen spektrum sayısı arasında önemli ölçüde fark yoksa ilgili moleköl "gecici" tanımlanır ve ilgili MOLFIT dosyasına yazılır. Nihai olarak her molekölün katkısını belirlemek için Otomatik izgi Tanıma yazılımının sonunda kullanılan overall.molfit dosyasına eklenir.

3.2.1.4. Tüm Moleköl Uydurma İşlemi

Tüm “Tekli moleköl uydurma” işlemi bittikten sonra AlgorithmXMLFileOverAll adlı giriş parametresi tarafından kontrol edilen tekli moleköl uydurma işlemi sonucunda tanımlanan tüm moleküller için eş zamanlı “Tüm moleköl uydurma” işlemi gerçekleştirilir. “Tekli Moleköl Uydurma” işlemindeki gibi “Tüm Moleköl Uydurma” işlemi kullanıcı tanımlı frekans aralıęı daha küçük aralıklara ayrılmadıęından ve iç içe gemiş izgiler hesaba katılacaęı için Moleköl ve bileşen sayısına baęlı olarak bir hayli zaman alan bir işlemdir. “Tüm Moleköl Uydurma” işlemi tamamlandıktan sonra, Otomatik izgi Tanıma yazılımı overall.molfit dosyasında açıklanan her molekölün katkısını belirler.

3.2.2. Otonom Gauss Ayırıştırma Yazılımı

Otonom Gauss Ayırıştırma (AGD-Autonomous Gaussian Decomposition), spektral verilerde bulunan Gauss bileşenlerini otomatik olarak ayırıştıran bir algoritmadır. AGD, türevlenebilir spektroskopik ve makine öğrenmesi teknikleri ile spektrum içerisinde bulunan bileşen sayısını ve bu bileşenlerin konumunu genişlięini ve genlięini tespit eder. AGD algoritması, Gauss

eğrilerinin ayrıştırılması sorununa en küçük karelerin en aza indirilmesi yöntemi ile çözüm bulmuştur. Bu işlem için elde edilecek olan parametrelerin başlangıç değerlerini tahmini olarak girilmesi gerekmektedir. Başlangıç tahminleri iyi yapılırsa spektrum ayrıştırma işleminin çoğu gerçekleştirilmiş olur ve en küçük kareler yöntemini kullanarak yapılan eğri uydurma işlemi çok hızlı bir şekilde optimum sonuca ulaşır.

$$G(x; \alpha, \mu, \sigma) = \alpha e^{-(x-\mu)^2/2\sigma^2} \quad (3.30)$$

Denklem 3.30'da x ve $f(x)$ sırası ile spektruma ait verilerin frekans birimini ve yoğunluk değerini temsil etmektedir. Gauss denkleminde bulunan diğer a , σ ve μ sembolleri ise her gauss eğrisi için sırası ile genlik, genişlik ve konum değerlerini temsil etmektedir.

3.2.2.1. Türevlenebilir Spektroskopi

Türevlenebilir spektroskopi, spektral veri hakkında bilgi sahibi olmak için spektral verilerin türevinin alınması yöntemine dayalı bir tekniktir. Görüntü içerisindeki gradyan, eğim ve kenar gibi şekilleri tespit edilmesine olanak sağladığı için bu teknik genelde bilgisayarda görü uygulamalarında kullanılmaktadır. AGD, spektrumda kaç tane gauss bileşeni olduğuna ve bu bileşenlerin konumunun nerede olduğuna türevlenebilir spektroskopi tekniği ile tespit eder.

AGD, spektral verideki her yerel minimum noktasını bir tahmin değeri olarak kabul eder ve bu değer aynı zamanda, $f(x)$ fonksiyonunun ikinci türevini alarak elde edilen $f(x)$ 'in eğrilik değeridir. Bu teknik ile iç içe geçmiş gauss eğrileri tespit edilir. Matematiksel olarak elde edilmesi gereken değerler aşağıdaki Denklem 3.31, 3.32, 3.33 ve 3.34 ile sağlanır.

$$f/\varepsilon_0 \quad (3.31)$$

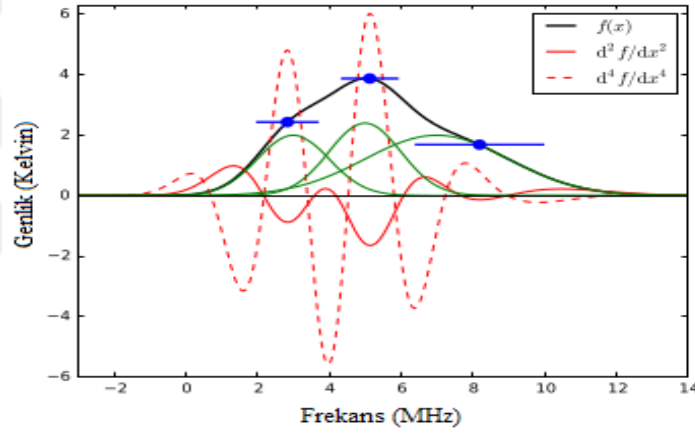
$$d^2 f/dx^2 < 0 \quad (3.32)$$

$$d^3 f/dx^3 = 0 \quad (3.33)$$

$$d^4 f/dx^4 > 0 \quad (3.34)$$

İdeal ortamda yani gürültü olmayan bir ortamda denklem 3.31'deki $\varepsilon_0 = 0$ olarak kabul edilir. Fakat gerçek verilerde gürültü olduğundan spektral verilerde çizgi oluşmasına sebep olur. Bu nedenle bu gürültü değerlerini elimine etmek için bir eşik değeri belirlenmelidir.

Denklem 3.32 eğimi negatif yaparken, Denklem 3.33 ve 3.34 ile eğimin yerel minimum noktası elde edilir yani eğimin konumu elde edilmiş olur. Denklem 3.31, 3.32, 3.33 ve 3.34'te bulunan x 'e sırası ile bileşenlerin konumunu belirten μ_n değeri $n \in \{1, 2, 3, 4, \dots\}$ değerleri için bileşenlerin tahmini konum değerlerini belirtir. Şekil 3.16, ideal ortamda (gürültünün olmadığı ortam) denklem 3.31-3.34 kullanılarak elde edilen bileşenlerin konumunu gösteren türev spektroskopisi örneği gösterilmektedir. Yeşil ve siyah çizgiler sırası ile 3 adet Gauss bileşenlerini ve toplam sinyali göstermektedir. 2. dereceden türev alınarak elde edilen değerler kırmızı çizgi ve 4. dereceden türev alınarak elde edilen değerler kırmızı kesik çizgiler ile belirtilmiştir. Verideki bileşenlerin konumu mavi daireler ile genişliği ise mavi çizgiler ile belirtilmiştir.



Şekil 3.16: Türevlenebilir Spektrum örneği.

Daha sonra AGD, bir bileşenin genişliği ile ikinci türevinin maksimumu arasındaki ilişkiyi kullanarak bileşenlerin genişliklerini tahmin eder. İzole edilmiş bir bileşen için, 2. türevin $x = \mu'$ noktasındaki tepe noktası Denklem 3.35 ile elde edilir.

$$\frac{d^2}{dx^2} G(x; \alpha, \mu, \sigma) l_{x=\mu} = -\frac{\alpha}{\sigma^2} \quad (3.35)$$

AGD, Denklem 3.35 ile elde edilen sonucu $\alpha \approx f(x)$ koşulu ile spektrum içerisindeki tüm bileşenlerin genişliğini hesaplamak için Denklem 3.36'da kullanılır.

$$\sigma_n^2 = f(x) \left(\frac{d^2 f(x)}{dx^2} \right) l_{x=\mu_n} \quad (3.36)$$

Son olarak AGD bileşenlerin genlik değerini α_n hesaplar. N adet bileşen için basitçe genlik tahminleri, bileşenlerin konumlarında oluşturulan orijinal verilerin değerleridir. Eğer, çok

sayıda iç içe geçmiş bileşen varsa, o zaman genlik değerleri önemli ölçüde fazla tahmin edebilir. AGD, çok fazla sayıda genlik tahmini yapılmış ise daha önce elde edilen konum ve genişlik tahminlerini kullanarak genlik tahminlerini tekrar-birleştir(de-blend) işlemi ile telafi eder. Denklem 3.31-3.34 kullanılarak $f(x)$ fonksiyonundaki bileşenleri tespiti doğru ve sorunsuz bir şekilde değişebilmelidir. Örneğin spektrumda bulunan herhangi bir gürültü sahte bileşen tahminlerine neden olmaktadır. Türevleri sonlu fark tekniklerini kullanarak hesaplamak, verideki gürültü oranını büyük ölçüde artırır. Özellikle dördüncü türev hesaplanırken elde edilen değerler anlamsızlaşır. Bu nedenle Tikhonov regularization tekniği kullanılarak farklılaşma düzenlenmiş olur.

$$R[u] = \alpha \int \sqrt{(D_x u)^2 + \beta^2} + \int |A_x u - f|^2 \quad (3.37)$$

Verinin düzenlenmiş türevi $u_{\min} = \arg_{\min}(R[u])$ şeklinde belirtilmiştir. Denklem 8’de, türev operatörü $D_x u = du/dx$ ve integral operatörü $A_x u = \int_{x_{\min}}^x u dx$ Denklem 3.37’nin ilk sağ terimi düzenleme terimidir. β sıfıra eşit olduğunda bu terim L1’e dönüşür. L1 aynı zamanda du/dx ’in normudur ve u ’yu parçalı değişmeze dönüştürür. $B > 0$ ise du/dx ‘in düzenleme terimi L2 normuna dönüşür ve u ’yu sorunsuz bir şekilde değişkenlik göstermesi için sınırlar. Türev için değişen çözümler üretmek amacı ile $\beta = 0.1$ olarak belirlenir ve türev bölme aralıkları yeniden ölçeklenir. Bu optimizasyon işlemi optimum sonuca ulaşana kadar tekrarlanır.

3.2.2.2. Makine Öğrenmesi Tekniği ile $\log \alpha$ Seçimi

Denetimli makine öğrenmesi tekniğinde, bilgisayara eğitim seti olarak adlandırılan giriş/çıkış çiftlerinden oluşan bir koleksiyon verilir ve ardından girdileri çıktılarla eşlemek için genel bir kural öğrenilir. Bu öğrenme süreci tamamlandığında, algoritma yeni giriş değerleri için çıkış değerlerini tahmin etmede öğrenilen bu kuralı kullanır.

AGD’yi eğitmek için denetimli makine öğrenimi tekniği kullanılmıştır ve bilinen ayrıştırımlara sahip eğitim amacı ile kullanılan bir spektral veride bileşen tahminlerinin doğruluğunu en üst düzeye çıkaran $\log \alpha$ ’nın optimal değeri denetimli makine öğrenmesi tekniği ile seçilmiştir. Eğitim seti, verinin bir alt kümesini el ile ayrıştırarak veya bileşenler yardımı ile yeni bir sentetik spektrum oluşturularak belirlenir.

Gerçek bileşenleri N_t içeren veriler üzerinde $\{\alpha_i^t, \mu_i^t, \sigma_i^t\}_{i=1}^{N_t} = t$ } AGD'nin sabit $\log \alpha$ ile çalıştırılmasıyla üretilen bileşen tahminleri $N_g \{\alpha_i^g, \mu_i^g, \sigma_i^g\}_{i=1}^{N_g} = g_\alpha$ verildiğinde, tahminlerin "doğruluğu" dengeli F-skoru kullanılarak tanımlanır. Dengeli F-skoru hem hassasiyete hem de geri çağırmağa bağlı olan bir sınıflandırma doğruluğu ölçüsüdür, böylece yanlış, eksik veya sahte olan bileşen tahminlerini elimine eder.

$$A(g_\alpha, t) = \frac{2N_c}{N_g + N_t} = A(\log \alpha) \quad (3.38)$$

Doğruluk oranı ise Denklem 3.38 ile belirlenmektedir. Bu denklemde N_c doğru tahminlerin sayısını belirtmektedir. Tahmin edilen tek bir bileşenin ($\alpha^g, \sigma^g, \mu^g$) genliği, konumu ve genişliğinin tümü aşağıdaki denklemlerle belirtilen sınırlar dahilindeyse, gerçek bir bileşenle ($\alpha^t, \sigma^t, \mu^t$) "doğru" eşleşme olarak kabul edilir:

$$c_1 < \frac{\alpha^g}{\alpha^t} < c_2 \quad (3.39)$$

$$\left| \frac{\mu^g - \mu^t}{\sigma^t} \right| < c_3 \quad (3.40)$$

$$c_4 < \frac{\sigma^g}{\sigma^t} < c_5 \quad (3.41)$$

Nihai çözüm, başlangıç genliklerine en az duyarlıdır, bu nedenle geniş bir aralık almak için c_1 ve c_2 (Denklem 3.39) değerleri seçilir. Bu aralık tahmin edilen genişliklere karşı daha hassastır, bu nedenle c_4 ve c_5 (Denklem 3.41) aralık değerleri daha dar seçilir. Son olarak, eğri uydurma işleminin en önemli parametresi olan konum değeri c_3 (Denklem 3.40) değeri ile eğitilmektedir. Tahmin edilen ve gerçek bileşenler arasındaki kısıtlamanın bire bir olması gerektiği ve dolayısı ile azalan genlik değerlerine göre karşılaştırma işleminin değerlendirilmesi gerçekleşmektedir.

$\log \alpha$ 'nın optimal değeri, AGD ile tahmin edilen bileşenler ve eğitilmiş veriden elde edilen gerçek sonuçlar arasındaki doğruluk oranını (Denklem 3.35) maksimum seviyeye çıkarır. Bu lineer olmayan optimizasyon süreci gradient descent (eğim iniş) yönteminin modifiye edilmiş versiyonu ile gerçekleştirilmiştir. |

4. BULGULAR

Bu bölümde, XCLASS'ın hesap yoğun kısımlarının GPU programlama tekniği ile hızlandırılması ve optimizasyon işlemlerinden sonra elde edilen sonuçlar verilecektir. XCLASS'ın CPU ve GPU olmak üzere her iki versiyonundaki çalışma süreleri, SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen veriler üzerinde test edilecek ayrıntılı sonuçlar sunulacaktır.

İlerleyen kısımlarda, Otomatik Çizgi Tanıma yazılımının ve Gauss paketi içerisinde bulunan AGD yazılımının SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen veriler üzerindeki spektral çizgilerin ve bu çizgilerin yükseklik, konum ve genlik gibi parametrelerinin tespiti işlemindeki sonuçları, başarı oranlarının karşılaştırılması ayrıntılı olarak verilecektir.

4.1. XCLASS'IN HIZLANDIRILMASI İŞLEMİNİN DENEYSEL SONUÇLARI

Tez kapsamında analizi yapılan ilk performans testi XCLASS'ın hesap yoğun kısımlarının GPU programlama tekniği ile hızlandırılması üzerinedir. Çalışma süresini ölçmek için XCLASS içerisinde bulunan fonksiyonlardan biri olan myXCLASSFit kullanılmıştır. myXCLASSFit fonksiyonun kullanılmasının sebebi çalıştığında hızlandırma işleminin yapıldığı bütün fonksiyonları çağırmasıdır. Fonksiyon, myXCLASS programı ile deneysel verileri eğri uydurma işleminden geçirmek için Levenberg-Marquardt algoritmasını kullanır. Fonksiyon, maksimum döngü sayısına ulaşıldığında veya ch^2 değeri 10^{-7} 'nin altına düştüğünde durmaktadır. Bu çalışmada, gözlemsel veriyi uydurmak için MOLFIT dosyasında bulunan "ArH+" isimli molekülün seçilmesinin sebebi daha önce MOLFIT dosyasında tanımlanmış olmasıdır. Gözlemsel veri olarak Sagittarius B2 (Sgr B2) adlı büyük moleküler gaz ve toz bulutundan 617200.0- 618000.0 MHz frekans aralığındaki spektral veriler kullanılmıştır. myXCLASSFit fonksiyonu sonlandığında, optimize edilmiş model spektrumunu gözlemsel verilerle birlikte gösteren bir spektrum oluşur. En sonunda modellenen spektrum ile gözlem verisi karşılaştırılır.

Test işlemlerinin gerçekleştirilmesinde kullanılan 2 adet Tesla K20m GPU kartının özellikleri Tablo 4.1'de verilmiştir. Model spektrumunun oluşturulduğu, DetectionEquation, InterpolateQGPU ve diğer fonksiyonların çağrıldığı ModelCalSpectrum adlı fonksiyonların CPU ve GPU üzerinde çalışma süreleri ve her bir fonksiyon için performans kazancı da yüzde

olarak Tablo 4.2’de gösterilmiştir. Bu fonksiyonların CPU ve GPU üzerinde çalışma sürelerine bakıldığında sırası ile %87, %14 ve %62 performans kazancı olduğu görülmektedir. Yapılan hızlandırma işleminin doğruluğunu test etmek için XCLASS içerisinde bulunan test_demo-from-manual, test_demo-from-manual_Genetic+LM+Error, test_dust-continuum-fit adlı test fonksiyonları kullanılmıştır. Bu test fonksiyonlarının her bir döngüsünün hem CPU hem de GPU platformu üzerinde çalışma süresi çalışma süresi hesaplanarak Tablo 4.3’te gösterilmiştir. Bu tablo sonuçlarına bakıldığında da sırasıyla %98, %98 ve %93’lük bir performans kazancı elde edildiği tespit edilmiştir. Deneysel sonuçlara bakıldığında, GPU programlama tekniği ile elde edilen bu performans kazancının, astronomi gibi çok büyük boyutlu veriler ile uğraşmak zorunda olan bilim dalı için önemli bir seviyede olduğu görülmektedir.

Tablo 4.1: Tesla K20m GPU ekran kartının özellikleri.

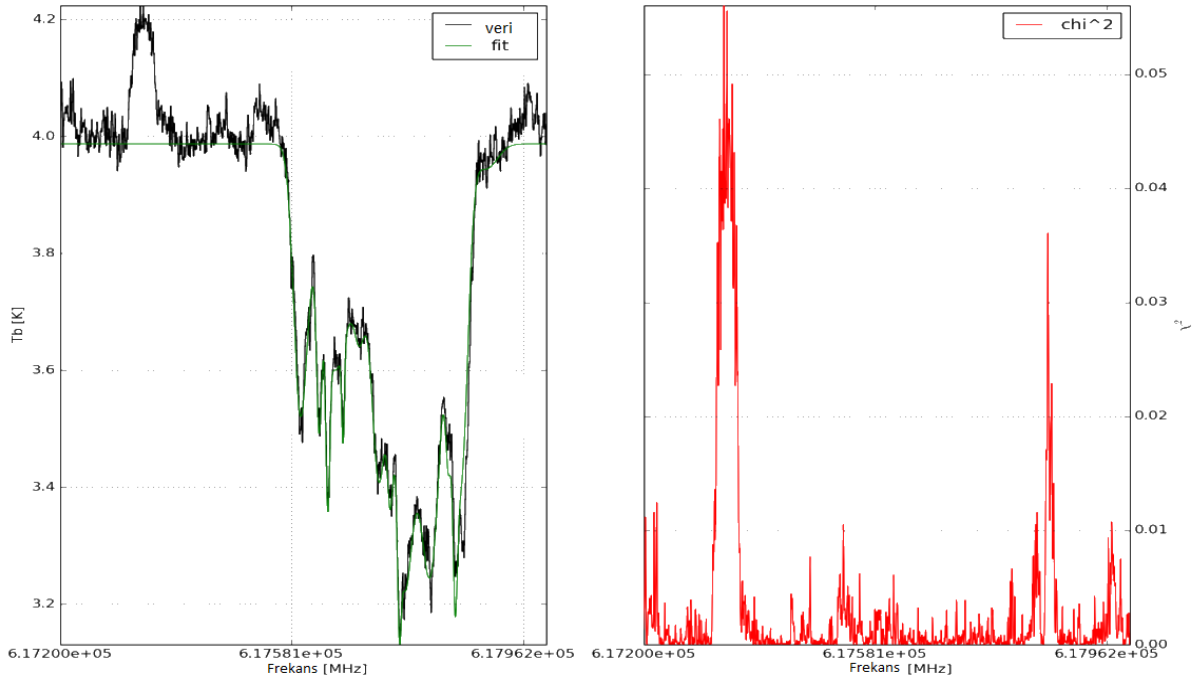
Özellik	Değer
Cihaz İsmi	Tesla K20m
Bellek Hızı (KHz)	2600000
Bellek Bant Genişliği (bits)	320
En yüksek Bellek Bant Genişliği (GB/s)	208
Maksimum Grid Boyutu	2147483647x65535x65535
Maksimum Blok Boyutu	1024x1024x64
Her Bloktaki Maksimum İş Parçacığı Sayısı	1024

Tablo 4.2: Deneysel sonuçlar ve performans kazancı.

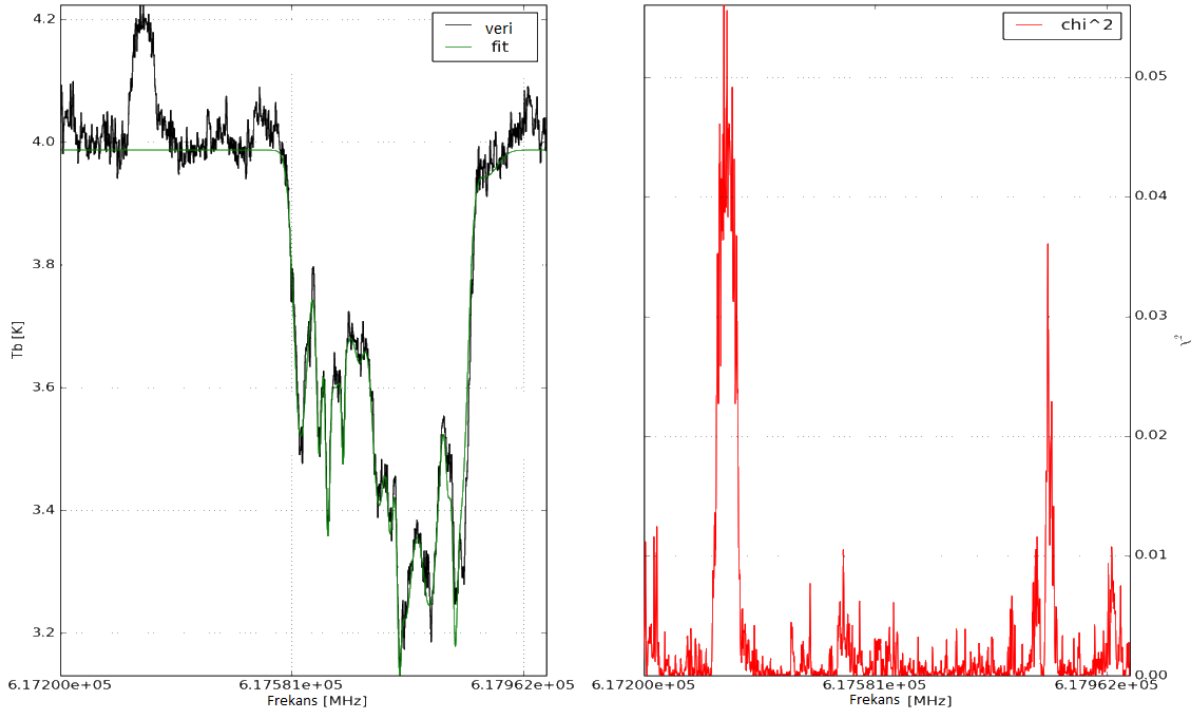
Fonksiyon İsmi	CPU’da çalışma süresi	GPU’da çalışma süresi	Performans Kazanımı
DetectionEquation	7,6293E-06 s	9,5367E-07 s	%87
InterpolateQGPU	9,1062E-05 s	7,7962E-05 s	%14
ModelCalcSpectrum	2,5320 s	0,9645 s	%62

Tablo 4.3: Diğer test fonksiyonlarının CPU ve GPU platformu üzerinde çalışma süreleri ve performans kazancı.

Test Fonksiyon ismi	CPU’da çalışma süresi	GPU’da çalışma süresi	Performans Kazanımı
test_demo-from-manual	5,7932 s	0,0943 s	%98
test_demo-from-manual_Genetic+LM+Error	5,8041 s	0,0892 s	%98
test_dust-continuum-fit	70,4631 s	4,2856 s	%93

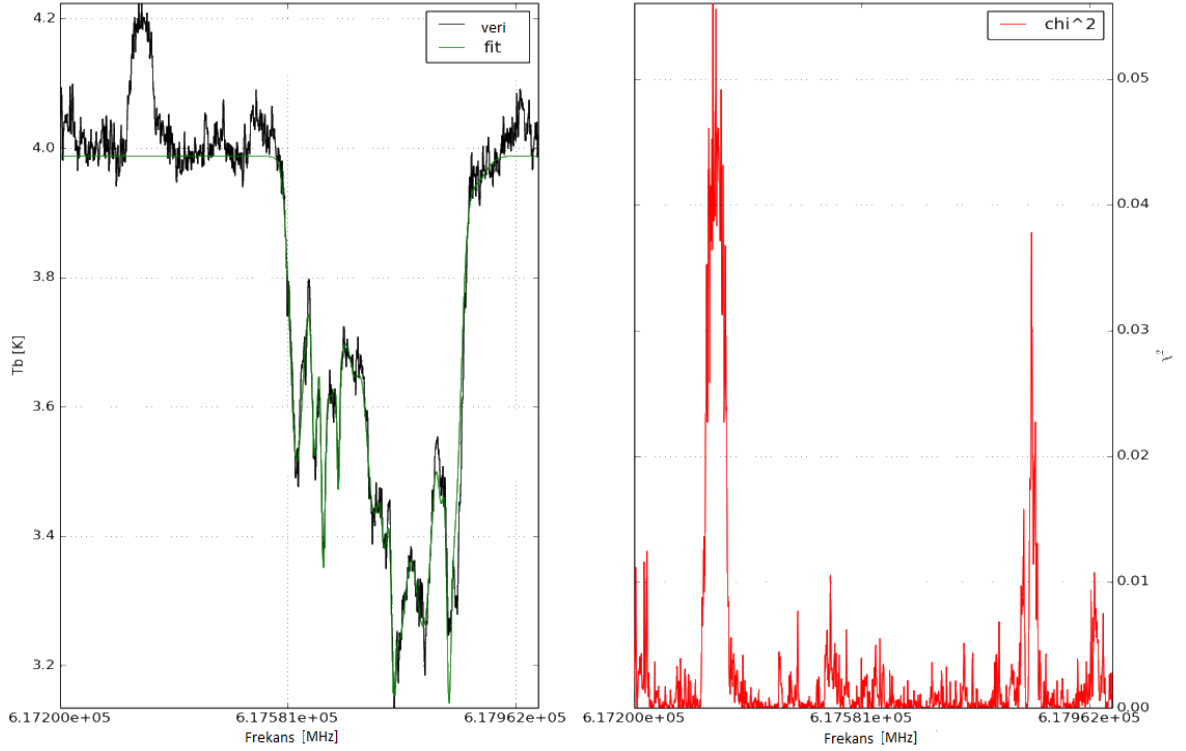


Şekil 4.1: CPU platformu üzerinde ArH + molekülünün eğri uydurma işlemi (solda) ve χ^2 hata vektörü (sağda).

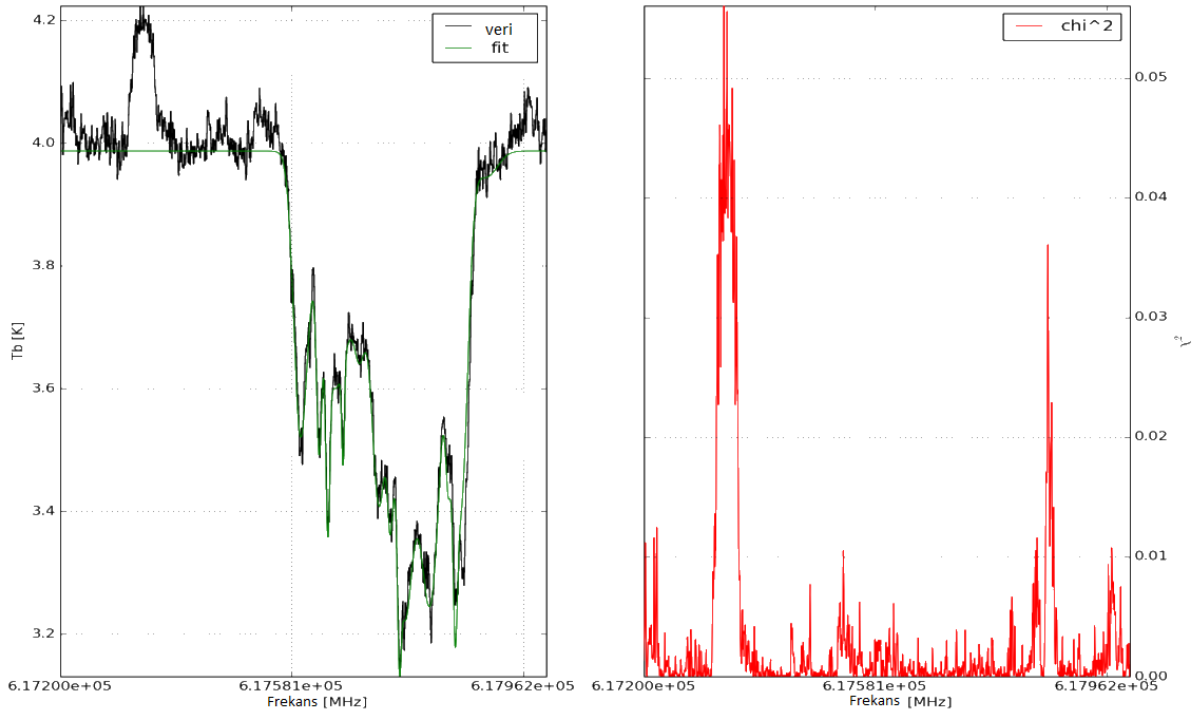


Şekil 4.2: GPU platformu üzerinde ArH + molekülünün eğri uydurma işlemi (solda) ve χ^2 hata vektörü (sağda).

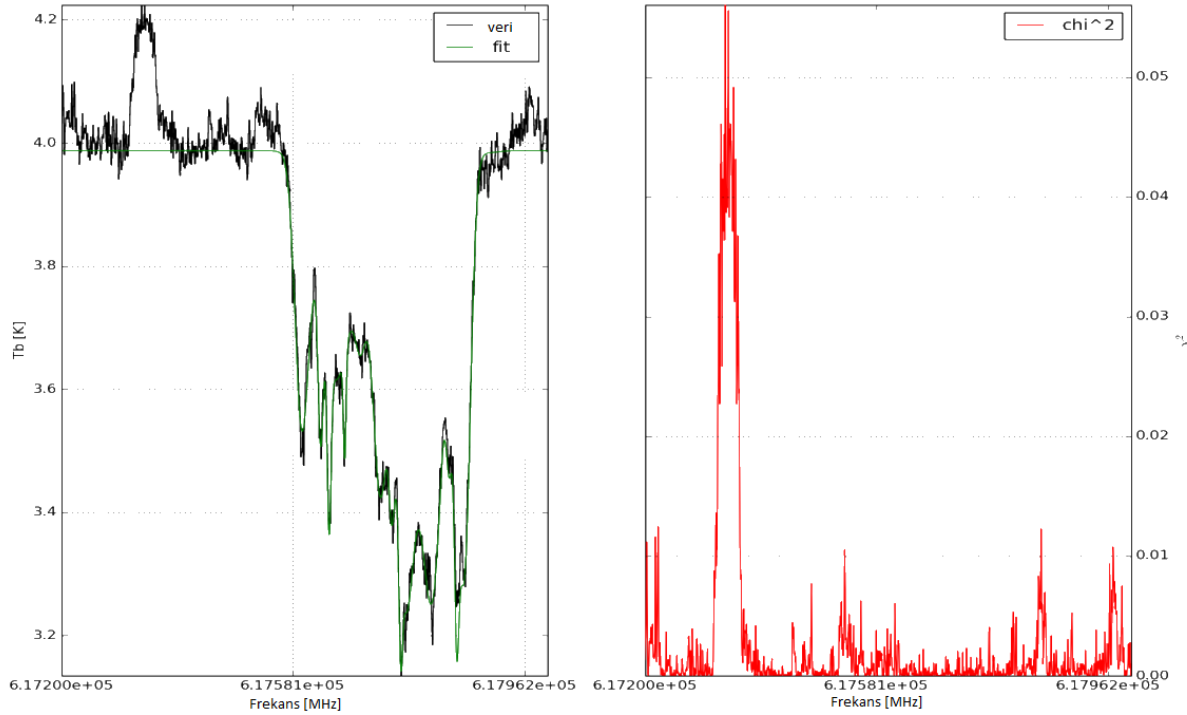
Şekil 4.1’de XCLASS’ın CPU üzerinde çalışan versiyonundan elde edilen gözlemsel veriler (siyah çizgi) ve modellenmiş spektrum (yeşil çizgi) ve eğri uydurma işlemine ait hata değerini gösteren χ^2 gösterilmektedir. Şekil 4.2’de GPU üzerinde çalışan versiyonundan elde edilen gözlemsel veriler, modellenmiş spektrum ve χ^2 değerleri gösterilmiştir. Şekil 4.1 ve Şekil 4.2 karşılaştırıldığında; CPU üzerinde yapılan spektrum analizinin, GPU tekniği ile gerçekleştirildiğinde de bozulmadığı; ayrıca GPU programlama tekniği kullanılarak uygulamanın daha hızlı bir şekilde gerçekleştirildiği görülmüştür.



Şekil 4.3: 10 döngü ile yapılan modelleme işlemi.



Şekil 4.4: 50 döngü ile yapılan modelleme işlemi.



Şekil 4.5: 100 döngü ile yapılan modelleme işlemi.

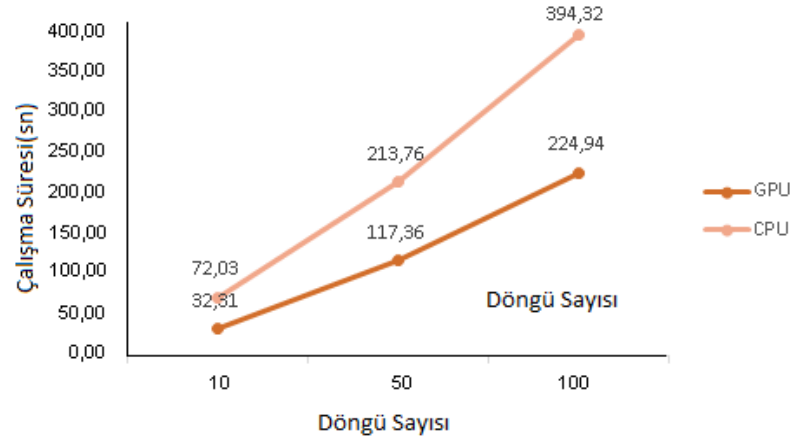
myXCLASSFit fonksiyonunu 672000 KB boyutundaki ArH^+ molekülü üzerinde test etmek ve döngü sayısının eğri uydurma işlemi üzerindeki başarı oranına ve çalışma süresine olan etkisini

tespit etmek için döngü sayısı sırası ile 10, 50 ve 100 olarak alınmıştır. Şekil 4.3, 4.4 ve 4.5'te sırası ile 10, 50 ve 100 döngü ile gerçekleştirilen myXCLASSFit fonksiyonunun sonuçları sunulmuştur. Elde edilen sonuçlar, döngü sayısı arttıkça eğri uydurma işleminin başarı oranının arttığını ve χ^2 hata değerinin azaldığını göstermektedir.

Tablo 4.4'te ise döngü sayısına bağlı olarak myXCLASSFit fonksiyonunun her iki platform üzerinde toplam çalışma süresinin değişimi ve performans kazanımı gösterilmiştir. Tablo 4.4'ten elde edilen verilerle Şekil 4.6'daki grafik, spektrum analizinin doğruluk oranını artırmak amacı ile döngü sayısı artırılması her iki platform üzerinde de çalışma süresinde artışa neden olduğunu göstermiştir. Şekil 4.6'daki grafik incelendiğinde döngü sayısının artması ile CPU platformu üzerinde çalışma süresinin daha hızla arttığı gözlemlenmiştir.

Tablo 4.4: GPU ve CPU platformu üzerinde döngü sayısının toplam çalışma süresine etkisi ve döngü sayısına bağlı olarak toplam çalışma süresinin artış oranı.

Döngü Sayısı	GPU'da çalışma süresi	CPU'da çalışma süresi	Performans Kazanımı
10 döngü	32,31 s	72,03 s	%55
50 döngü	117,36 s	213,76 s	%45
100 döngü	224,94 s	394,32 s	%42



Şekil 4.6: GPU ve CPU platformu üzerinde döngü sayısının toplam çalışma süresine etkisinin grafiksel gösterimi.

4.2. SPEKTRAL ÇİZGİLERİN TESPİTİ İŞLEMİNİN SONUÇLARI

Tez kapsamında analizi yapılan ikinci çalışma, SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral verilerde bulunan iç içe geçmiş, tespit edilmesi zor spektral çizgilerin tespiti işlemi üzerinedir. Bu amaçla, Bölüm 3.2.1’de ayrıntılı olarak anlatılan Otomatik Çizgi Tanıma Yazılımı geliştirilmiştir. Otomatik Çizgi Tanıma Yazılımını test etmek için kullanılan spektral veriler, Gauss paketi içerisinde bulunan Otonom Gauss Ayırıştırma yöntemi ile birlikte spektral çizgilerin tespiti işleminde kullanılmıştır. Bu bölümde her iki algoritma ile elde edilen sonuçlar karşılaştırılmış ve başarı oranları ayrıntılı olarak sunulmuştur.

4.2.1. Otomatik Çizgi Tanıma Yazılım’ının Test İşlemleri ve Sonuçları

Spektral verilerin tespit edilmesi zor iç içe geçmiş, tespit edilmesi zor spektral çizgilerin tespiti işlemi amacı ile geliştirilen Otomatik Çizgi Tanıma Yazılımı, belirlenen frekans aralığında laboratuvar ortamında elde edilen özellikleri ile herhangi molekül veya atoma ait spektral çizgilerin tespiti işlemini gerçekleştirmiştir. Bu yazılımın test işleminde kullanılan parametreler, moleküller ve elde edilen sonuçları bir sonraki bölümlerde adım adım sunulmuştur.

4.2.2. Otomatik Çizgi Tanıma Yazılımı Giriş Parametre Değerleri

Noise: Spektrumdaki gürültü değerini belirtir. Bu değerden düşük olan tüm çizgiler göz ardı edilir. Kelvin cinsinden yazılır.

MaxOverestimationHeight: tekli molekül uydurma işleminden sonra oluşan spektrum için maksimum tahmin değerini % cinsinden belirtir. Bu değer varsayılan olarak 10 atanır. Ayrıca Tüm molekül uydurma işleminde kullanılan MaxOverestimationHeight parametre değerinde farklıdır.

Eğer model spektrum, gözlemsel veriden fazla değilse “gecici” olarak tanımlanır ve o moleküle ait MOLFIT dosyası Otomatik Çizgi Tanıma Yazılımının son aşamasında her molekülün katkısını belirleme amacı ile kullanılan overall.molfit dosyasına eklenir.

DefaultMolfitFile: Tekli molekül uydurma işleminde kullanılacak olan molfit dosyasının adını ve yolunu belirtir. Eğri uydurma işlemi sırasında, default.molfit dosyasında tanımlanan (ilk) molekülün adı, mevcut molekülün adıyla değiştirilir.

EstimateParamFlag (T/F): Tekli molekül uydurma işleminde için gerçekleştirilen Parametre tahmin algoritmasının kullanılıp kullanılmadığını gösteren bayrak değişkenidir.

VelBin(km/s): Histogram analizinde hesap yoğunluğu azalmak için hızı bölmelere ayırmada kullanılır.

ElowMax: Kelvin cinsinden tanımlanan bu parametre ile belirlenen eşik değerinden daha yüksek değere sahip hız bileşenlerinin geçiş frekanslarını dahil edilmemesi imkânı sunar.

Tolerance: Tekli molekül uydurma işleminde tahmin edilen maksimum çizgi sayısını belirler. Eğer elde edilen sonuç Tolerance değerinden düşük ise molekül “gecici” olarak tanımlanır ve overall.molfit dosyasına eklenir.

TypeSelection: Bu parametreye bağlı olarak, parametre tahmin algoritması emisyon çizgilerinin mi (TypeSelection = " em ") ya da soğurma çizgilerinin mi (TypeSelection = " abs ") veya her ikisinin mi (TypeSelection = " all ") analizini yapacağına karar verir.

MaxNumTransInFit: Parametre Tahmin Algoritmasında tespit edilen maksimum geçiş sayısını belirtir (varsayılan: 1.e99).

SelectedMolecules: Otomatik Çizgi Tanıma Yazılımı verilen frekans aralığında bulunan veri tabanındaki tüm moleküller için yazılmıştır. Tüm moleküller için yazılımın çalışma süresi çok uzun zaman aldığından Otomatik Çizgi Tanıma Yazılımı tarafından dikkate alınması gereken moleküllerin listesinin tanımlandığı listedir.

Seçilen moleküllerin Listesi: ["HCCCN", "CH3OH", "C2H5OH", "CH3CN", "SO", "SO2"]

StrongMoleculeList: Verilen spektrumda bulunma olasılığı yüksek olan moleküllerin listesini tutar.

MinColumnDensityEmis: Tek bir molekülün bir bileşeninin minimum kolon yoğunluğu (emisyon çizgileri için) belirtir. Overall.molfit dosyasına eklenir.

MinColumnDensityAbs: Tek bir molekülün bir bileşeninin minimum kolon yoğunluğu (soğurma çizgileri için) belirtir. Overall.molfit dosyasına eklenir.

NumberIteration: Otomatik Çizgi Tanıma Yazılımının maksimum yenileme sayısını belirtir (varsayılan olarak 50 atanır).

experimentalData: Bu parametre, deneysel verileri Otomatik Çizgi Tanıma yazılımına göndermek için dosyasının yolunu ve adını tanımlar. Bu dosyadaki ilk sütun MHz cinsinden frekansı, ikinci sütun bir spektrumun ışın sıcaklığını (yoğunluğunu) tanımlar.

TelescopeSize: Teleskobun boyutunu belirler (varsayılan: 1).

Inter_Flag (T/F): Tek çanak (F) veya interferometrik verilerin (T) kullanılıp kullanılmadığını belirten bayrak değişkenidir (varsayılan: F).

tBack: Arka plan sıcaklık değerini Kelvin cinsinden belirtir (varsayılan: 0).

tslope: Sıcaklık eğim değerini belirtir (varsayılan: 0).

N_H: Hidrojen sütun yoğunluğu değerini cm^{-2} cinsinden belirtir (varsayılan: 0).

beta_dust: Toz için spektral indeks değerini belirtir (varsayılan: 0).

kappa_1300: Toz kütlesi opaklığı değerini $\text{cm}^2 \text{g}^{-1}$ cinsinden belirtir (varsayılan: 0.01).

4.2.3. Test İşleminde Kullanılan Spektral Veri ve Veri Tabanı

Otomatik Çizgi Tanıma Yazılımını test etmek için SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen veriler kullanılmıştır. Bu veriler "SyntheticData.dat" adlı dosyada 2 sütundan oluşmaktadır. 1. sütun frekans 2. sütun ise sıcaklık değerini gösterir.

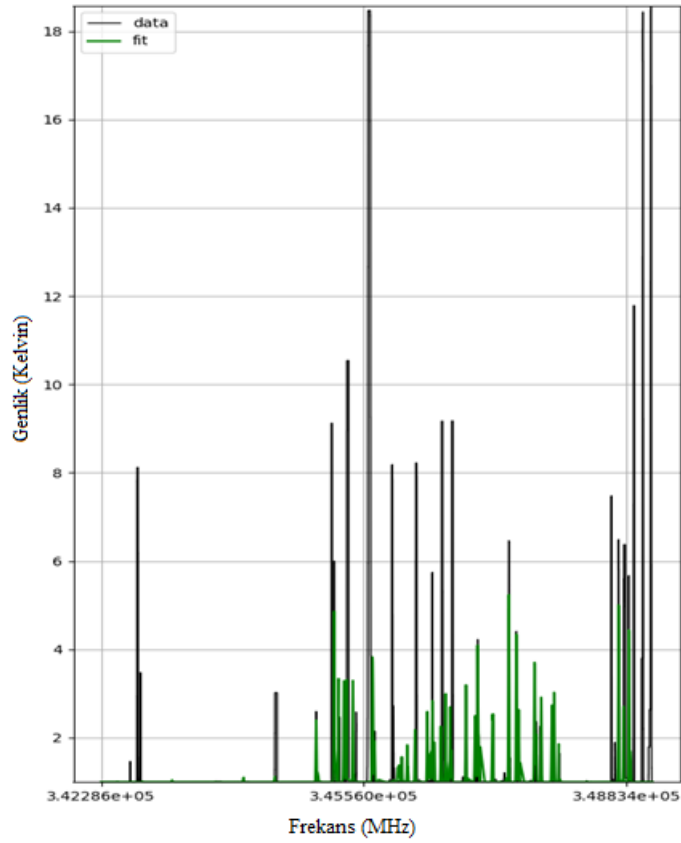
Veri tabanı: "cdms_sqlite.db" adlı 2.5GB boyutundaki veri tabanı kullanılmıştır. Bu veri tabanında her molekül için ayrılan her sütunda aşağıdaki bilgiler bulunmaktadır.

1. Sütun: Molekül ismi
2. Sütun: Frekans (MHz)
3. Sütun: Yoğunluk ($\text{nm}^2 \text{MHz}$)
4. Sütun: Einstein A katsayısı (s^{-1})

5. Sütun: Belirsizlik (MHz)
6. Sütun: Düşük Enerji (cm^{-1})
7. Sütun: Üst düzey bozunum
8. Sütun: Nükleer spin izomer
9. Sütun: HFS
10. Sütun: Yüksek düzey kuantum sayıları
11. Sütun: Düşük düzey kuantum sayıları

4.2.3.1. “ $\text{C}_2\text{H}_5\text{OH}$ ” molekülü için Sonuçlar

342281.000000-349243.000000 MHz olarak belirlenen frekans aralığında otomatik çizgi tanıma yazılımı ile “ $\text{C}_2\text{H}_5\text{OH}$ ” adlı moleküle ait toplam 51 adet spektral çizgi tespit edilmiştir. Şekil 4.7’de tespit edilen spektral çizgiler grafik üzerinde gösterilmiştir. Siyah çizgiler SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral verileri, yeşil ile gösterilen çizgiler otomatik çizgi tanıma yazılımı tarafından tespit edilen spektral çizgileri temsil etmektedir. Tablo 4.5’te otomatik çizgi tanıma yazılımı ile $\text{C}_2\text{H}_5\text{OH}$ molekülüne ait tespit edilen spektral çizgilerin bulunduğu frekans aralığı, konumu ve genlik değerleri ayrıntılı olarak sunulmuştur.



Şekil 4.7: Otomatik Çizgi Tanıma Yazılımı ile " C_2H_5OH " molekülüne ait tespit edilen spektral çizgilerin grafiksel gösterimi.

Tablo 4.5: Otomatik Çizgi Tanıma Yazılımı ile " C_2H_5OH " molekülüne ait tespit edilen spektral çizgilerin frekans aralığı, konum ve genlik değerleri.

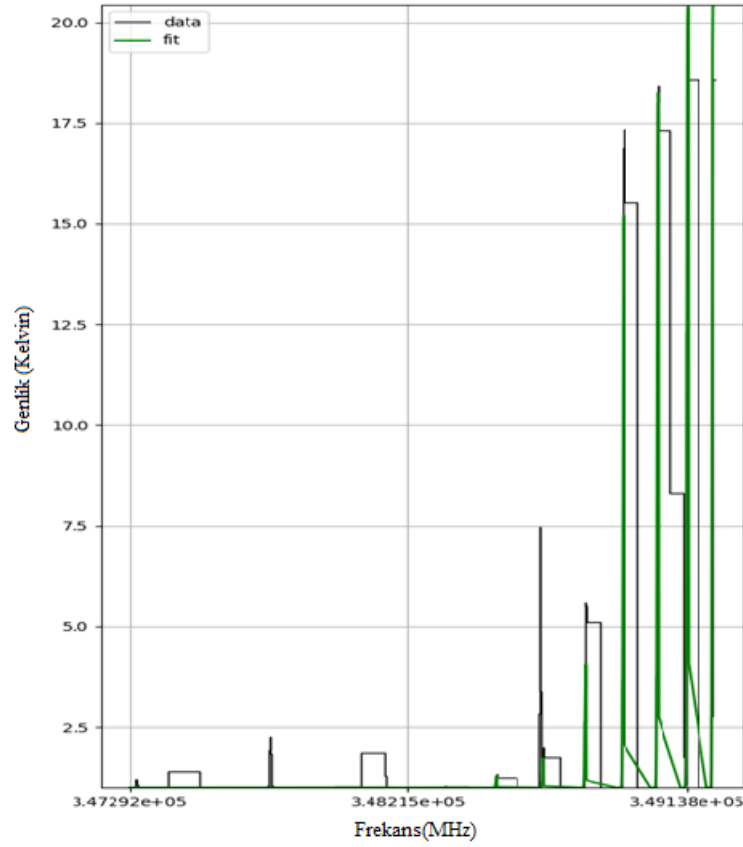
Hız Aralığı	Frekans Aralığı (MHz)	Frekans (MHz)	Genlik (Kelvin)
31	344940.156100 – 344985.399700	344968.0	1.3986703604956352
35	345163.949300 – 345197.61380	345192.0	3.8603122247758748
		345195.0	3.1149637445123393
36	345219.248000 – 345247.971930	345245.0	2.160153453199373
37	345277.696700 – 3453512.964800	345306.0	0.6447798365400621
		345313.0	2.286164032916152
		345345.0	0.91236020377539
		345349.0	2.121305864138381
38	345398.165100 – 3454392.262260	345426.0	2.293788741439411
41	345584.708100 – 3457072.898850	345667.0	2.677560074568778

		345675.0	2.448285902175116
		345696.0	0.922441001596122
45	345948.043200 – 3460713.747080	346035.0	0.567084106774502
46	346075.563100 – 3461112.350680	346104.0	0.8368024416618289
47	346118.158800 – 3462843.555190	346201.0	1.180120677653849
48	346308.042200 – 3464526.442190	346352.0	1.5909173937526289
		346380.0	0.610625407957424
		346402.0	0.6859745162995441
		346417.0	1.839881833026317
		346443.0	0.896087178443582
49	346495.346900 – 3465268.793570	346523.0	1.258965648908744
		346524.0	1.2495451486766989
50	346536.084800 – 3465833.162750	346581.0	1.837508163903395
51	346597.204100 – 3466506.436560	346638.0	1.6984879222931122
55	346806.583300 – 3468565.956030	346835.0	2.194949975643663
56	346918.900100 – 3470142.554540	346947.0	1.4617749110456288
		346957.0	1.451947235639596
		346981.0	3.0925476126089446
		346993.0	1.0132294497003609
58	347103.346900 – 3471798.120140	347165.0	1.5019388778298581
		347175.0	1.531107895708756
		347176.0	1.5180009715327998
		347177.0	1.455094450151873
61	347340.929000 – 3474216.463160	347369.0	4.237965231194325
62	347435.523000 – 3475119.566850	347464.0	3.341589860373049
		347492.0	1.6351218323680738
65	347660.065000 – 3477341.335720	347688.0	2.433751089350395
		347694.0	2.304740140409811
		47711.0	1.0778562882907372

66	347744.903800 – 3478130.242500	347773.0	1.9131784878581088
		347777.0	1.302023557296391
67	347860.761600 – 3479926.331360	347906.0	1.735261635173066
		347908.0	1.5102952071046452
		347934.0	2.024977252740614
		347990.0	0.7713503113277209
77	348676.450300 – 3487757.387230	348732.0	1.671818994164497
		348738.0	4.003259743833248
		348739.0	3.965164543733291
78	348776.868000 – 3488049.011300	348802.0	1.520389595195566
79	348837.046400 – 3488804.225350	348865.0	3.440921057342834
		348866.0	3.4034173312066773

4.2.3.2. “CH₃CN” Molekülü için Sonuçlar

342281.000000-349243.000000 MHz olarak belirlenen frekans aralığında otomatik çizgi tanıma yazılımı ile "CH₃CN" adlı moleküle ait toplam 6 adet spektral çizgi tespit edilmiştir. Şekil 4.8’de tespit edilen spektral çizgiler grafik üzerinde gösterilmiştir. Siyah çizgiler SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral verileri, yeşil ile gösterilen çizgiler otomatik çizgi tanıma yazılımı tarafından tespit edilen spektral çizgileri temsil etmektedir. Tablo 4.6’da otomatik çizgi tanıma yazılımı ile CH₃CN molekülüne ait tespit edilen spektral çizgilerin bulunduğu frekans aralığı, konumu ve genlik değerleri ayrıntılı olarak sunulmuştur.



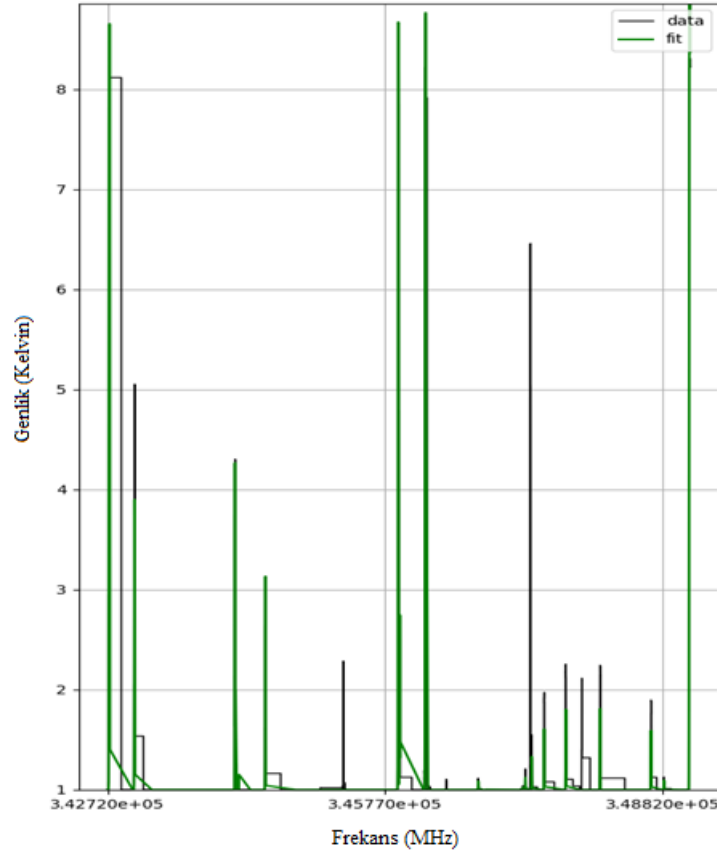
Şekil 4.8: Otomatik Çizgi Tanıma Yazılımı ile " CH_3CN " molekülüne ait tespit edilen spektral çizgilerin grafiksel gösterimi.

Tablo 4.6: Otomatik Çizgi Tanıma Yazılımı ile " CH_3CN " molekülüne ait tespit edilen spektral çizgilerin frekans aralığı, konum ve genlik değerleri.

Hız Aralığı	Frekans Aralığı (MHz)	Frekans (MHz)	Genlik (Kelvin)
8	348634.690900 – 348662.716679	348660.0	0.7299354017824169
9	348774.624500 – 348802.657514	348800.0	3.0591701725240146
10	348901.401200 – 348929.440769	348927.0	14.1998626094998
11	349014.970500 – 349043.015941	349040.0	17.2494570335252
12	349115.287100 – 349143.337727	349141.0	19.38773982061548
13	349202.310600 – 349230.365727	349228.0	19.43084695766864

4.2.3.3. “CH₃OH” Molekölü için Sonuçlar

342281.000000-349243.000000 MHz olarak belirlenen frekans aralığında otomatik çizgi tanıma yazılımı ile "CH₃OH" adlı moleküle ait toplam 13 adet spektral çizgi tespit edilmiştir. Şekil 4.9’da tespit edilen spektral çizgiler grafik üzerinde gösterilmiştir. Siyah çizgiler SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral verileri, yeşil ile gösterilen çizgiler otomatik çizgi tanıma yazılımı tarafından tespit edilen spektral çizgileri temsil etmektedir. Tablo 4.7’de otomatik çizgi tanıma yazılımı ile CH₃OH molekülüne ait tespit edilen spektral çizgilerin bulunduğu frekans aralığı, konumu ve genlik değerleri ayrıntılı olarak sunulmuştur.



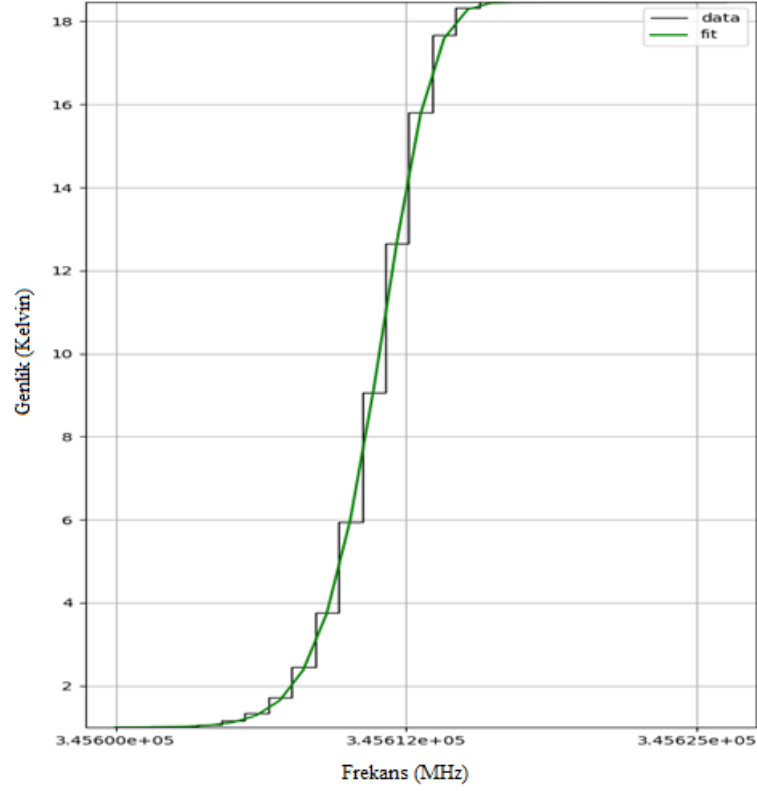
Şekil 4.9: Otomatik Çizgi Tanıma yazılımı ile "CH₃OH" molekülüne ait tespit edilen spektral çizgilerin grafiksel gösterimi.

Tablo 4.7: Otomatik Çizgi Tanıma Yazılımı ile "CH₃OH" molekülüne ait tespit edilen spektral çizgilerin frekans aralığı, konum ve genlik değerleri.

Hız Aralığı	Frekans Aralığı (MHz)	Frekans (MHz)	Genlik (Kelvin)
1	342719.830000 - 342747.549967	342742.0	7.653636720198197
2	342995.882000 - 343023.616239	343018.0	2.900049910153952
4	344099.360000 - 344127.151292	344122.0	3.2666157314132667
7	344433.469000 - 344461.277566	344456.0	2.132476847458846
10	345894.110000 - 345937.232873	345916.0	7.671147247887346
		345932.0	1.7478702306010572
11	346192.776000 - 346222.268609	346215.0	7.764287442855089
		346217.0	7.764351107074702
20	347497.451000 - 347525.417981	347520.0	0.606604630157396
21	347735.442000 - 347763.421286	347758.0	0.803479100824668
23	348112.547000 - 348140.545783	348135.0	0.8110592390030411
24	348671.951000 - 348699.978706	348694.0	0.5919989946557409
27	349097.020000 - 349125.069683	349119.0	7.856655617708006

4.2.3.4. "HCCCN" Molekülü için Sonuçlar

Belirlenen frekans aralığında otomatik çizgi tanıma yazılımı ile "HCCCN" adlı moleküle ait tespit edilen toplam spektral çizgi sayısı sadece 1'dir. Şekil 4.10'da tespit edilen spektral çizgi grafik üzerinde gösterilmiştir. Siyah çizgiler SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral verileri, yeşil ile gösterilen çizgi otomatik çizgi tanıma yazılımı tarafından tespit edilen spektral çizgiyi temsil etmektedir. Tablo 4.8'de otomatik çizgi tanıma yazılımı ile HCCCN molekülüne ait tespit edilen spektral çizginin bulunduğu frekans aralığı, konumu ve genlik değeri ayrıntılı olarak sunulmuştur.



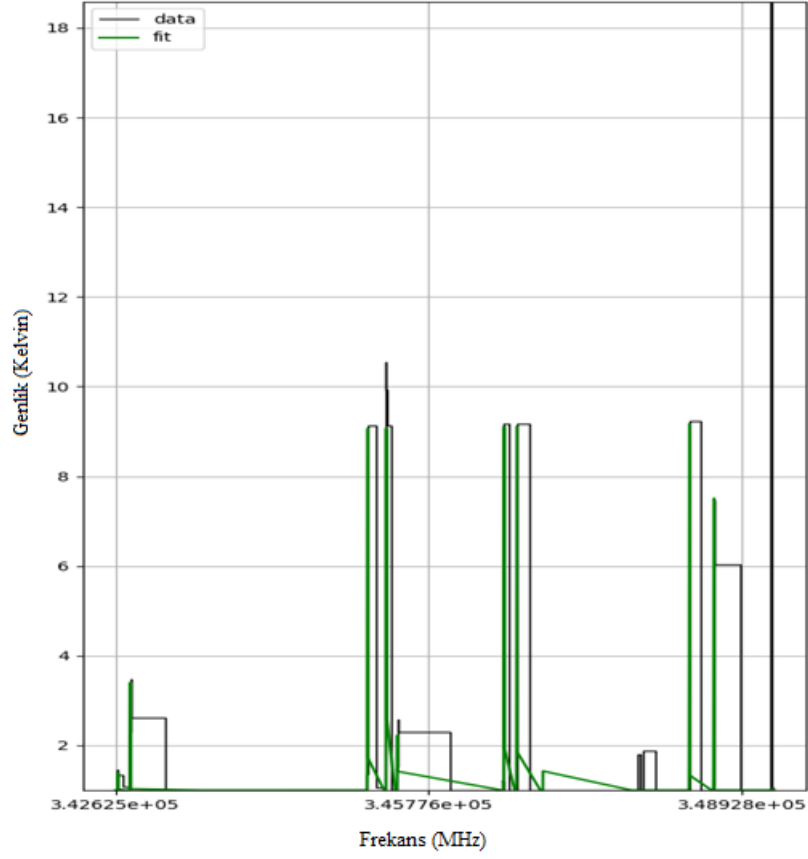
Şekil 4.10: Otomatik Çizgi Tanıma Yazılımı ile "HCCCN" molekülüne ait tespit edilen spektral çizginin grafiksel gösterimi.

Tablo 4.8: Otomatik Çizgi Tanıma Yazılımı ile "HCCCN" molekülüne ait tespit edilen spektral çizginin frekans aralığı, konum ve genlik değeri.

Hız Aralığı	Frekans Aralığı (MHz)	Frekans (MHz)	Genlik (Kelvin)
1	345599.010000 - 345626.878827	345624.0	17.46484821169929

4.2.3.5. "SO₂" Molekülü için Sonuçlar

342281.000000-349243.000000 MHz olarak belirlenen frekans aralığında otomatik çizgi tanıma yazılımı ile "SO₂" adlı moleküle ait toplam 8 adet spektral çizgi tespit edilmiştir. Şekil 4.11'de tespit edilen spektral çizgiler grafik üzerinde gösterilmiştir. Siyah çizgiler SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral verileri, yeşil ile gösterilen çizgiler otomatik çizgi tanıma yazılımı tarafından tespit edilen spektral çizgileri temsil etmektedir. Tablo 4.9'da otomatik çizgi tanıma yazılımı ile SO₂ molekülüne ait tespit edilen spektral çizgilerin bulunduğu frekans aralığı, konumu ve genlik değerleri ayrıntılı olarak sunulmuştur.



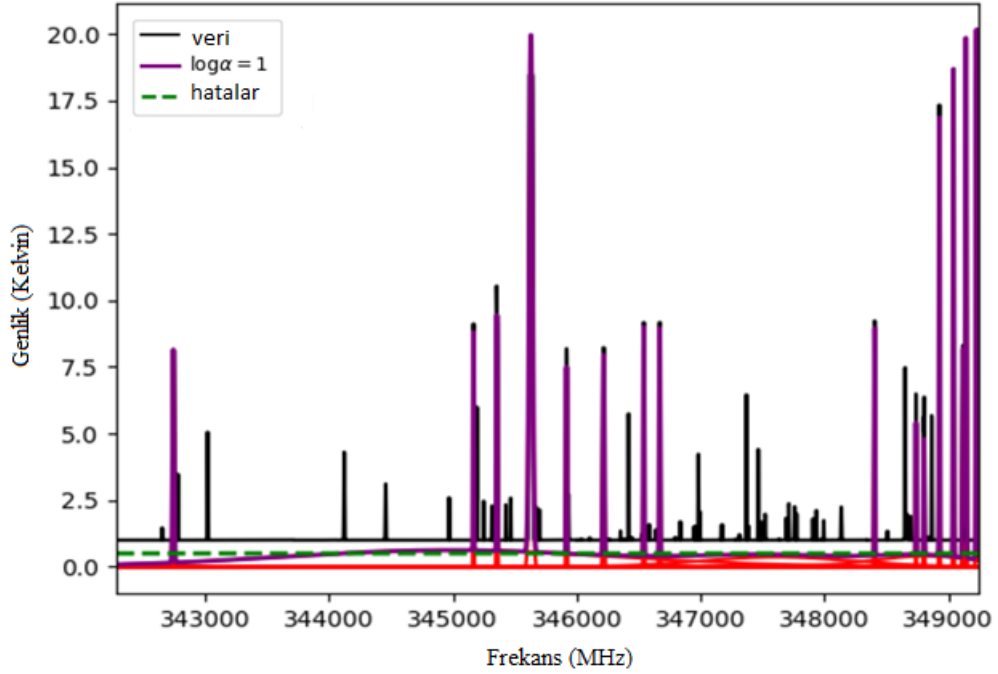
Şekil 4.11: Otomatik Çizgi Tanıma Yazılımı ile "SO₂" molekülüne ait tespit edilen spektral çizgilerin grafiksel gösterimi.

Tablo 4.9: Otomatik Çizgi Tanıma Yazılımı ile "SO₂" molekülüne ait tespit edilen spektral çizgilerin frekans aralığı, konum ve genlik değeri.

Hız Aralığı	Frekans Aralığı (MHz)	Frekans (MHz)	Genlik (Kelvin)
2	342751.625400 - 342779.347011	342777.0	1.531184456884291
4	345138.970800 - 345166.815842	345164.0	8.059920221107523
5	345328.537700 - 345356.632656	345354.0	8.066294492125117
6	345438.984100 - 345466.844654	345464.0	0.674664468712999
7	346513.878400 - 346541.794528	346539.0	8.105813014029778
8	346642.169100 - 346670.091861	346668.0	8.110131420153692
13	348377.800000 - 348405.812497	348403.0	8.167917359800775
14	348623.353100 - 348651.378293	348649.0	4.925186141181196

4.2.3. AGD Yazılımının Sonuçları

Gerçekleştirilen Otomatik Çizgi Tanıma Yazılımının başarı oranını belirlemek amacı ile Gauss paketi içerisinde bulunan AGD Yazılımı ile karşılaştırmalar yapılmıştır. SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral veriler üzerinde, C_2H_5OH , CH_3CN , CH_3OH , SO , $HCCCN$, SO_2 moleküllerine ait spektral çizgilerin tespiti işlemi AGD yazılımı kullanılarak toplamda 22 adet spektral çizgi tespit edilmiştir. (Şekil 4.12 Tablo 4.10) Her bir moleküle ait tespit edilen spektral veriler ve bu çizgilerin parametreleri bir sonraki bölümlerde ayrıntılı olarak sunulmuştur.



Şekil 4.12: AGD yazılımı ile SgrB2 adlı bölgede tespit edilen tüm spektral çizgiler.

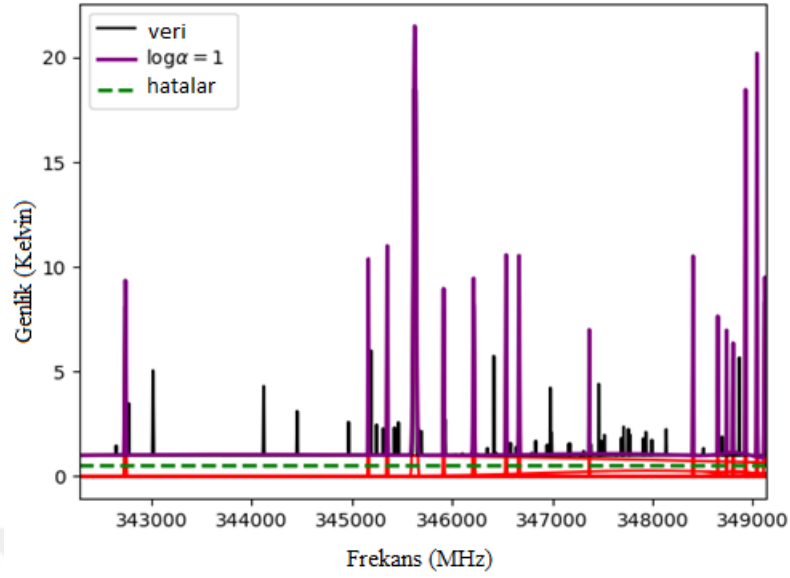
Tablo 4.10: AGD yazılımı ile SgrB2 adlı bölgede tespit edilen tüm spektral çizgilerin ait FWHM, frekans ve genlik değerleri.

FWHM	Frekans (MHz)	Genlik (Kelvin)
11.5907	349228	19.8999
12.1416	349119	7.87574
-29.4875	345629	19.4176
3138.45	344973	0.622581
-65.9557	345495	9.89393e-06
7.58415	349041	18.4189

6.63424	348927	16.4531
8.74995	345354	8.8978
6.61866	348404	8.53218
8.65921	346668	8.59578
8.42048	346540	8.60625
6.33941	345165	8.20054
9.28265	349141	19.5276
12.7404	346216	7.52047
8.1242	345916	7.06873
13.9428	342742	7.99692
605.337	348551	0.011057
-4.62173	348738	5.01525
1403.86	347614	0.365047
832.482	346612	0.021057
1011.64	348820	0.40254
8.20756	348803	4.3616

4.2.3.1. “C₂H₅OH” Molekülü için Sonuçlar

AGD yazılımı ile SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral veriler üzerinde C₂H₅OH molekülüne ait 20 adet spektral çizgi tespit edilmiştir. Elde edilen spektral çizgiler Şekil 4.13 ‘te grafiksel olarak gösterilmiştir. Şekilde siyah renkli çizgi ile gösterilen çizgiler gözlem yolu ile elde edilen verileri, mor renk ile gösterilen çizgiler ise AGD yazılımı ile tespit edilen çizgileri belirtmektedir. Kırmızı renkli çizgi, tespit edilen spektral çizginin frekans noktasını belirtmektedir. Yeşil renkli noktalı çizgiler ise belirlenen gürültü seviyesini işaret etmektedir. Tablo 4.11’de ise tespit edilen C₂H₅OH adlı molekülün her bir spektral çizgisine ait FWHM, frekans ve genlik değerleri ayrıntılı olarak sunulmuştur.



Şekil 4.13: AGD yazılımı ile “C₂H₅OH” adlı moleküle ait spektral çizgiler.

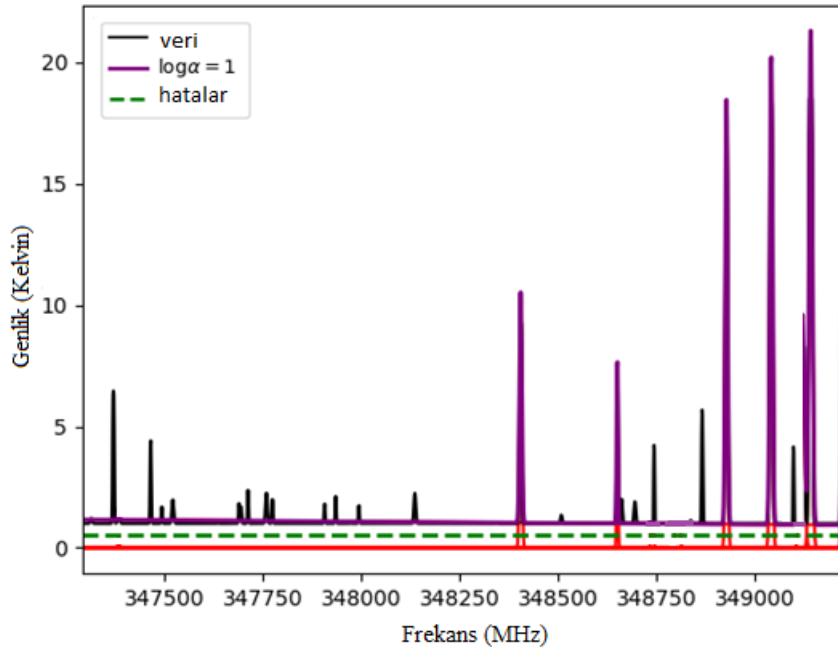
Tablo 4.11: AGD yazılımı ile “C₂H₅OH” adlı moleküle ait tespit edilen her bir spektral çizgiye ait FWHM, frekans ve genlik değerleri.

FWHM	Frekans (MHz)	Genlik (Kelvin)
-29.6802	345629	20.5152
12800	343904	1.04009
14.7095	345524	0.00144589
7.58644	349041	19.3976
6.63292	348927	17.422
8.9338	345354	10.0477
6.66312	348404	9.53457
8.66466	346668	9.54893
8.42295	346540	9.55653
6.51235	345165	9.37074
12.5073	349120	8.7015
12.7221	346216	8.45065
8.16633	345916	8.03684
12.7313	342742	8.34208
3.80699	348650	6.7829

-4.6615	348738	5.97125
4.41654	347369	5.92982
2353.95	347959	0.268408
373.845	348836	0.244514
8.22577	348803	5.27687

4.2.3.2. “CH₃CN” Molekölü için Sonuçlar

AGD yazılımı ile SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral veriler üzerinde CH₃CN molekülüne ait 6 adet spektral çizgi tespit edilmiştir. Elde edilen spektral çizgiler Şekil 4.14 ‘te grafiksel olarak gösterilmiştir. Şekilde siyah renkli çizgi ile gösterilen çizgiler gözlem yolu ile elde edilen verileri, mor renk ile gösterilen çizgiler ise AGD yazılımı ile tespit edilen çizgileri belirtmektedir. Kırmızı renkli çizgi, tespit edilen spektral çizginin frekans noktasını belirtmektedir. Yeşil renkli noktalı çizgiler is belirlenen gürültü seviyesini işaret etmektedir. Tablo 4.12’de ise tespit edilen CH₃CN adlı molekülün her bir spektral çizgisine ait FWHM, frekans ve genlik değerleri ayrıntılı olarak sunulmuştur.



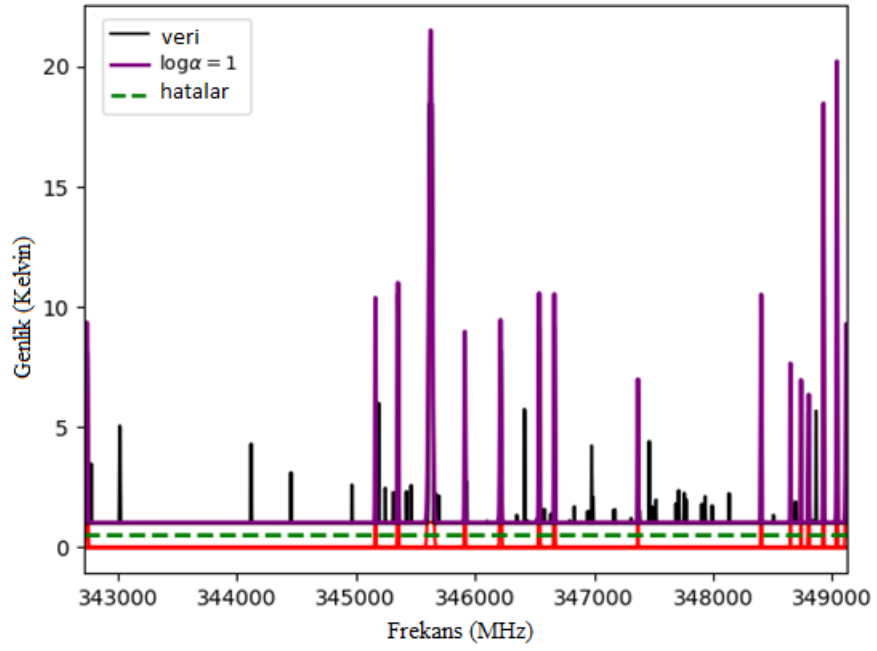
Şekil 4.14: AGD yazılımı ile “CH₃CN” adlı moleküle ait spektral çizgiler.

Tablo 4.12: AGD yazılımı ile “CH₃CN” adlı moleküle ait tespit edilen her bir spektral çizgiye ait FWHM, frekans ve genlik değerleri.

FWHM	Frekans (MHz)	Genlik (Kelvin)
7.56994	349041	19.367
6.67201	348927	17.4894
-9.21529	349141	20.3557
6.63457	348404	9.49938
8.71156	349226	19.9205
11.9961	349119	8.68999

4.2.3.3. “CH₃OH” Molekülü için Sonuçlar

AGD yazılımı ile SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral veriler üzerinde CH₃OH molekülüne ait 17 adet spektral çizgi tespit edilmiştir. Elde edilen spektral çizgiler Şekil 4.15 ‘te grafiksel olarak gösterilmiştir. Şekilde siyah renkli çizgi ile gösterilen çizgiler gözlem yolu ile elde edilen verileri, mor renk ile gösterilen çizgiler ise AGD yazılımı ile tespit edilen çizgileri belirtmektedir. Kırmızı renkli çizgi, tespit edilen spektral çizginin frekans noktasını belirtmektedir. Yeşil renkli noktalı çizgiler is belirlenen gürültü seviyesini işaret etmektedir. Tablo 4.13’te ise tespit edilen CH₃OH adlı molekülün her bir spektral çizgisine ait FWHM, frekans ve genlik değerleri ayrıntılı olarak sunulmuştur.



Şekil 4.15: AGD yazılımı ile “CH₃OH” adlı moleküle ait spektral çizgiler.

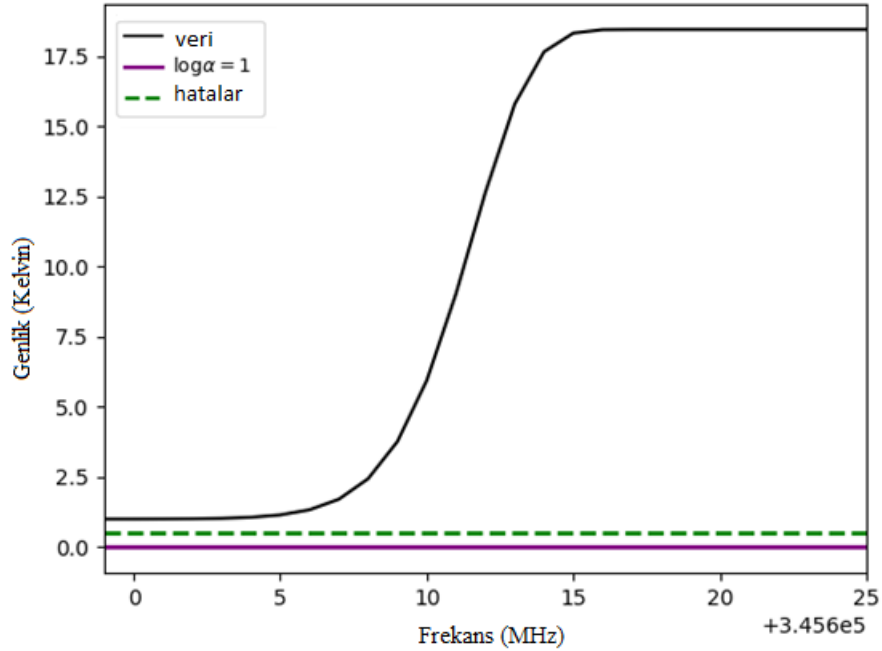
Tablo 4.13: AGD yazılımı ile “CH₃OH” adlı moleküle ait tespit edilen her bir spektral çizgiye ait FWHM, frekans ve genlik değerleri.

FWHM	Frekans (MHz)	Genlik (Kelvin)
-29.6431	345629	20.4918
-0.0169337	345519	0.15334
7.53424	349041	19.3002
6.64051	348927	17.4354
8.91704	345354	10.0299
6.62732	348404	9.49037
8.66145	346668	9.54579
8.41315	346540	9.54639
6.50247	345165	9.35839
12.6848	346216	8.42841
8.13317	345916	8.01306
12.7146	342742	8.3319
-4.4501	347369	5.96278
-4.71745	348738	6.02499

3.81229	348650	6.78852
12.6831	349120	8.29247
8.36147	348803	5.35019

4.2.3.4. “HCCCN” Molekülü için Sonuçlar

AGD yazılımı ile SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral veriler üzerinde HCCCN molekülüne ait spektral çizgi tespit edilememiştir. Şekil 4.16 ‘da siyah renkli çizgi ile elde edilen gözlemsel veride HCCCN molekülüne ait bir adet spektral çizgi olmasına rağmen AGD yazılımı ile bu spektral çizgi tespit edilememiştir.

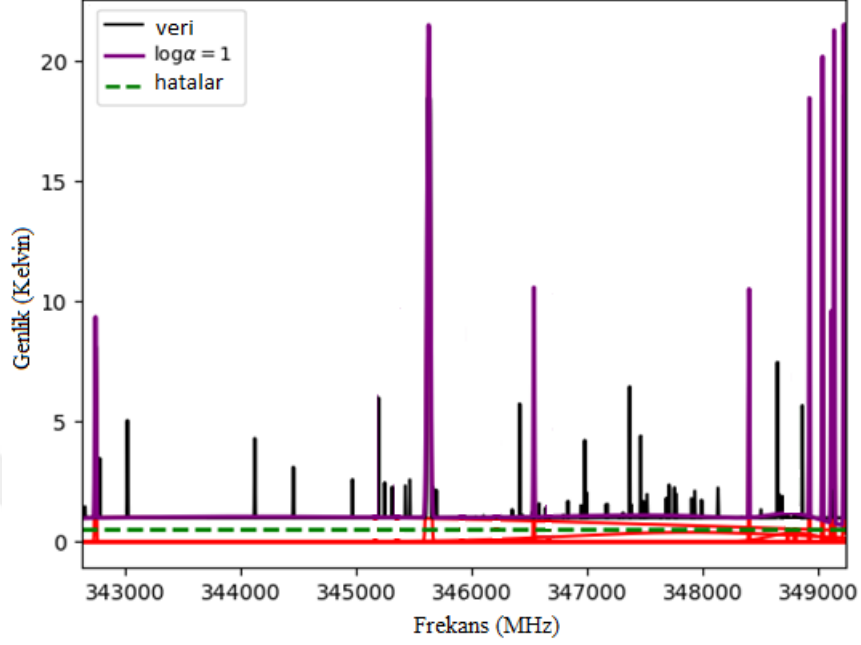


Şekil 4.16: AGD yazılımı ile “HCCCN” adlı moleküle ait tespit işleminin grafiksel gösterimi.

4.2.3.5. “SO₂” Molekülü için Sonuçlar

AGD yazılımı ile SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen spektral veriler üzerinde SO₂ molekülüne ait 8 adet spektral çizgi tespit edilmiştir. Elde edilen spektral çizgiler Şekil 4.17 ‘de grafiksel olarak gösterilmiştir. Şekilde siyah renkli çizgi ile gösterilen çizgiler gözlem yolu ile elde edilen verileri, mor renk ile gösterilen çizgiler ise AGD yazılımı ile tespit edilen çizgileri belirtmektedir. Kırmızı renkli çizgi, tespit edilen spektral çizginin frekans noktasını belirtmektedir. Yeşil renkli noktalı çizgiler ise belirlenen gürültü seviyesini

işaret etmektedir. Tablo 4.14’te ise tespit edilen SO₂ adlı molekülün her bir spektral çizgisine ait FWHM, frekans ve genlik değerleri ayrıntılı olarak sunulmuştur.



Şekil 4.17: AGD yazılımı ile “SO₂” adlı moleküle ait spektral çizgiler.

Tablo 4.14: AGD yazılımı ile “SO₂” adlı moleküle ait tespit edilen her bir spektral çizgiye ait FWHM, frekans ve genlik değerleri.

FWHM	Frekans (MHz)	Genlik (Kelvin)
11.5757	349228	20.8557
-29.6817	345629	20.5157
-1.17827	345447	1.04631e-05
7.59955	349041	19.4214
6.63112	348927	17.4191
8.93357	345354	10.0469
8.41283	346540	9.54802
9.29264	349141	20.5267

4.2.3. Otomatik Çizgi Tanıma Yazılımı ve AGD Algoritmalarının Karşılaştırılması

SgrB2 adlı büyük kütleli yıldız oluşum bölgesinden elde edilen verilerde C₂H₅OH, CH₃CN, CH₃OH, HCCCN, SO ve SO₂ moleküllerine ait spektral çizgiler ve bu çizgilere ait genlik, frekans ve konum değerlerinin tespiti işlemi geliştirilmiştir. Otomatik Çizgi Tanıma algoritması

ile toplamda 79 adet AGD yazılımı ile toplamda 22 adet spektral çizgi tespit edilmiştir. Her iki algoritma ile SO molekülüne ait herhangi bir spektral çizgi tespit edilememiştir.

Tablo 4.15: Otomatik Çizgi Tanıma ve AGD algoritmaları ile her bir moleküle ait ve toplam tespit edilen spektral çizgilerin sayısı.

Molekül	Otomatik Çizgi Tanıma	AGD
C ₂ H ₅ OH	51	20
CH ₃ CN	6	6
CH ₃ OH	13	17
HCCCN	1	0
SO	0	0
SO ₂	8	8
Toplam tespit edilen spektral çizgi sayısı	79	22

Tablo 4.15'te her iki algoritma ile her moleküle ait tespit edilen spektral çizgi sayısı ve toplam çizgi sayısı gösterilmiştir. Otomatik Çizgi Tanıma algoritması ile her bir moleküle ait tespit edilen çizgilerin sayısının toplamı toplam tespit edilen çizgi sayısına eşit olmasına rağmen AGD algoritmasında bu durum geçerli olmadığı görülmüştür. Bunun nedeni, Otomatik Çizgi Tanıma algoritması spektral çizgi tespit işlemini gerçekleştirirken her moleküle ait laboratuvar ortamında elde edilmiş moleküllere ait bilgiler içeren MOLFİT uzantılı dosyayı kullanmasıdır. Moleküller atomlara göre daha karmaşık bir yapıya sahiptir. Aynı frekans değerine sahip fakat farklı genlikte spektral çizgiler oluşur. Bu durum spektrumda iç içe geçmiş çizgiler oluşmasına neden olur. MOLFİT dosyası sayesinde farklı moleküllere ait aynı frekansta fakat farklı genlik değerlerine sahip spektral çizgiler Otomatik Çizgi Tanıma algoritması ile ayırt edilebilmiştir. AGD algoritması, aynı frekans fakat farklı genlik değerine sahip spektral çizgileri tek bir çizgi olarak algılamıştır. Bu durum tespit edilen herhangi bir spektral çizginin birden fazla moleküle aitmiş gibi görünmesine sebep olmuştur. Böylece Otomatik Çizgi Tanıma algoritması ile aynı konumda bulunan fakat farklı genlik değerine sahip, diğer bir ifade ile iç içe geçmiş tespit edilmesi zor spektral çizgilerin tespiti işleminde daha başarılı sonuçlar elde edilmiştir. |

5. TARTIŞMA VE SONUÇ

ALMA projesi vasıtası ile elde edilen veriler kullanılarak gerçekleştirilen ilk astroinformatik çalışması olma özelliği taşıyan bu tez çalışması kapsamında GPU programlama tekniği ile Köln Üniversitesi 1. Fizik Enstitüsü öğretim üyesi Prof. Dr. Peter SCHILKE tarafından geliştirilen, astronomik verileri modelleme amacı ile kullanılan XCLASS adlı yazılımının hesap yoğun kısımlarının hızlandırılması işlemi ve gözlem yolu ile elde edilen spektral verilerde iç içe geçmiş, tespit edilmesi zor ve XCLASS adlı yazılıma entegre edilmek üzere spektral çizgilerinin tespit edilmesini sağlayan Otomatik Çizgi Tanıma Yazılımı gerçekleştirilmiştir. Bu çalışmalar için gerçekleştirilen işlemler sırasıyla aşağıda verilmiştir.

Tez çalışmasının birinci aşamasında ilk olarak XCLASS'ın yapısı, hangi platformlarda çalıştığı, astronomik verileri modelleme işlemini hangi fonksiyonları kullanarak gerçekleştirdiği, fonksiyonlar çalışırken hangi dosyaların ve verilerin kullanıldığı ayrıntılı olarak incelenmiştir. XCLASS'ın astronomik verileri modelleme işlemini gerçekleştirirken fonksiyonların ortak olarak kullandığı diğer bir deyişle XCLASS'ın hesap-yoğun kısımları tespit edilip, bu kısımların GPU programlama tekniği kullanılarak hızlandırılması için uygun olup olmadığına karar verilmiştir.

XCLASS içerisinde bulunan myXCLASS adlı bölümde astronomik verileri modelleme işlemini gerçekleştiren Module_myXCLASS__Core.f90 ve Module_myXCLASS__GPU.f90 adlı program dosyalarında GPU programlama tekniği kullanılarak hızlandırma işlemi gerçekleştirilmiştir. Hızlandırma işlemi kapsamında Module_myXCLASS__Core.f90 adlı dosyada bulunan bütün parametre vektörleri için ayrı ayrı çağrılarak bir model spektrum oluşturan ve gözlem yolu ile elde edilen spektrum ile model spektrum arasındaki hata oranını hesaplayan, ModelCalcSpectrum adlı fonksiyon tarafından çağrılan InterpolateQGPU DetectionEquation adlı fonksiyonlar hızlandırılarak Module_myXCLASS__GPU.f90 adlı dosyaya yerleştirilmiştir. Hızlandırma işleminin başarı oran, ALMA adlı radyo teleskoplar aracılığı ile SgrB2 adlı büyük kütleli yıldız oluşum bölgelerinden elde edilen spektral veriler kullanılarak, iki adet Tesla K20 GPU kart üzerinde gerçekleştirilmiştir.

Gözlem yoluyla elde edilen çok sayıda bağımsız veri noktaları arasında bağlantı kurmak işlemini gerçekleştiren InterpolateQGPU adlı fonksiyonunun her bir iterasyonunun çalışma süresi $9,1062E-05$ sn'den $7,7962E-05$ sn'ye düşürülerek %14 performans kazancı sağlanmıştır.

Nesnenin sıcaklık, kimyasal bileşimi, hareketi, kökeni ve evrimi ve diğer ipuçları gibi pek çok parametrenin ortaya çıkarılması amacı ile belirli bir frekans aralığında ve bir boyutta izotermal nesne için ışımsal transfer denklemini çözerek bir spektrum modelleme işlemini gerçekleştiren DetectionEquation adlı fonksiyonunun her bir iterasyonunun çalışma süresi $7,6293E-06$ sn'den $9,5367E-07$ sn'ye düşürülerek %87 performans kazancı sağlanmıştır. Tüm parametre vektörleri için ayrı ayrı çağrılarak bir model spektrum oluşturan ve gözlem yolu ile elde edilen spektrum ile model spektrum arasındaki hata oranını hesaplayan, ModelCalcSpectrum adlı fonksiyonunun her bir iterasyonunun çalışma süresi $2,5320$ sn'den 1.44 sn'ye düşürülerek toplamda %43 performans kazancı sağlanmıştır.

GPU kartlar birçok bellek türüne sahiptir. Etkili bir programlama için bu bellek türlerinin verimli bir şekilde kullanılması gerekir. Eğer, sunucu ve cihaz arasında çok fazla transfer gerektiriyorsa, en iyi uygulama performansı için, mümkün olduğunca sunucu cihaz arasındaki veri transferlerini en aza indirmek ve aynı zamanda bu tür transferler gerektiğinde optimize edildiğinden emin olmak önemlidir. ModelCalcSpectrum adlı fonksiyonun çalışması anında gereksiz bellek transferleri kaldırılıp, optimizasyon işleminde geçirilerek toplam çalışma süresi 1.44 sn'den $0,9645$ sn'ye düşürülerek toplamda %62 performans kazancı sağlanmıştır.

XCLASS'ın modelleme işleminde başarı oranını artırmak için iterasyon sayısını artırmak gerekmektedir. İterasyon sayısının çalışma süresindeki etkisini ölçmek amacı ile iterasyon sayısı sırası ile 10, 50 ve 100 olarak alınmıştır. Belirlenen 10, 50 ve 100 iterasyon ile modelleme işleminin çalışma süresi 10 iterasyonda $72,03$ sn'den $32,31$ sn'ye düşürülerek %55 performans kazancı, 50 iterasyonda $213,76$ sn'den $117,36$ sn'ye düşürülerek %45 performans kazancı, 100 iterasyonda $394,32$ sn'den $224,94$ sn'ye düşürülerek %42 performans kazancı sağlanmıştır. Döngü sayısının spektrum analizinde hata değerinin değişimi ve işlem süresine olan etkisi incelendiğinde döngü sayısı arttıkça hata değerinin azaldığı ancak işlem süresinin arttığı görülmüştür. GPU üzerinde elde edilen sonuçlar ise işlem süresi üzerinde önemli bir performans kazancı elde edilebileceğini göstermiştir.

Ayrıca gerçekleştirilen hızlandırma işleminin başarı oranını XCLASS'ın diğer fonksiyonlarında ölçmek amacı ile XCLASS içerisinde bulunan test__demo-from-manual, test__demo-from-manual__Genetic+LM+Error ve test__dust-continuum-fit adlı fonksiyonları kullanılarak çalışma süresinde sırası ile $5,7932$ sn'den $0,0943$ sn'ye düşürülerek %98, $5,8041$ sn'den $0,0892$ sn'ye düşürülerek %98 ve $70,4631$ sn'den $4,2856$ sn'ye düşürülerek

%93 performans kazancı sağlanmıştır. Tüm fonksiyonlarda GPU'nun CPU'ya göre önemli bir performans kazancı sağladığı görülmüştür.

Bunun yanı sıra bu hızlandırma çalışmaları esnasında, astronomik çalışmalarda önem arz eden spektrum analizlerinin de GPU ve CPU'da aynı sonuçları verdiği, diğer bir deyişle verilerde herhangi bir bozulma olmadan hızlandırma işlemlerinin gerçekleştirildiği tespit edilmiştir. Bu tez çalışmasında gerçekleştirilen hızlandırma işlemleri sayesinde, çok daha büyük boyutlu astronomik verilerin daha kısa sürelerde işlenmesi ve daha fazla gök cisminin başarılı bir şekilde incelenmesi mümkün hale gelebilecektir.

Gelecek çalışmalarda; elde edilen sonuçları daha da iyileştirebilmek için daha fazla sayıda ya da daha üstün özelliklere grafik kartları kullanılması mümkündür. Ayrıca, spektrum hesaplaması için, GPU programlama tekniği kullanılarak yeni algoritmalar geliştirilebilir ve fonksiyonların daha da hızlandırılması için çalışmalar yapılabilir.

Tez çalışmasının ikinci aşamasında, XCLASS adlı yazılıma entegre edilmek üzere spektral verilerin içerisinde her bir moleküle ait çizgi sayısının tespiti işlemini gerçekleştiren, daha önce mevcut olmayan Otomatik Çizgi Tanıma yazılımı gerçekleştirilmiştir. Gerçekleştirilen yazılımı test etmek için ALMA projesinin radyo teleskopları ile elde edilen SgrB2 adlı büyük kütleli yıldız oluşum bölgesinin spektral verileri kullanılmıştır. Daha sonra karşılaştırma yapmak amacı ile Gauss paketi içerisinde bulunan iç içe geçmiş Gauss eğrilerinin tek bir Gauss eğrisi olarak algılayan AGD yazılımını ile aynı veriler üzerinde her bir moleküle ait ve toplam spektral çizgi sayısı tespiti işlemi gerçekleştirilmiştir. Otomatik Çizgi tanıma yazılımı ile belirlenen frekans aralığında C_2H_5OH , CH_3CN , CH_3OH , $HCCCN$ ve SO_2 moleküllerine ait sırası ile 51, 6, 13, 1 ve 8 adet olmak üzere toplamda 79 adet spektral çizgi tespit edilmiştir. SO molekülüne ait spektral çizgi tespit edilememiştir. AGD algoritması ile belirlenen frekans aralığında C_2H_5OH , CH_3CN , CH_3OH ve SO_2 moleküllerine ait sırası ile 20, 6, 17 ve 8 adet spektral çizgi tespit edilmiştir. $HCCCN$ ve SO moleküllerine ait spektral çizgi tespit edilememiştir. Fakat AGD algoritması ile spektral verilerin tamamında tespit edilen spektral çizgi sayısı 22'dir.

Otomatik Çizgi Tanıma yazılımının başarılı olmasının sebebi, her bir molekülün frekans genlik gibi laboratuvar ortamında elde edilen bilgileri kullanılarak tespit işlemi gerçekleştirilmesidir fakat AGD algoritmasında spektral çizgilerin tespiti işlemi gerçekleştirilirken moleküllere ait sadece bulunma olasılığı yüksek olan frekans aralığı kullanılmaktadır. Bu nedenle aynı frekans

aralığında fakat farklı genlik, FWHM değerine sahip birden fazla molekül bulunabilir. Bu nedenle her bir moleküle ait tespit edilen spektral çizgilerde çakışmalar meydana gelmiştir ve elde edilen sonuçlar Otomatik Çizgi Tanıma yazılımına göre daha az güvenilir ve az sayıda olmuştur.

Ayrıca spektral veriler doğrusal olmamaları sebebiyle, modellenmesi zor verilerdir. Bu nedenle spektrumun içerisindeki iç içe geçmiş Gauss eğrilerinin tespit edilmesi de zordur. AGD algoritması iç içe geçmiş Gauss eğrilerinin tek bir Gauss eğrisi olarak algıladığı için, tespit edilen spektral çizgi sayısı Otomatik Çizgi Tanıma yazılımına kıyasla daha az sayıda olduğu görülmüştür. Tespit edilemeyen her Gauss eğrisi, gözlem yapılan nesnenin bileşimi (hangi atom veya molekülden oluştuğu), sıcaklığı, hızı, yoğunluğu, bulunduğu ortamın manyetik alanı ve basıncı gibi bilgilerin tam olarak elde edilememesi anlamına gelmektedir. Gözlem yapılan bölgeler, kimyasal özelliklerine bakarak sınıflandırılabilirdiğinden, bir bölgede bir molekülün doğru bir şekilde belirlenmesi, büyük önem arz etmektedir. Bütün bunlardan anlaşılabileceği üzere iç içe geçmiş Gauss eğrilerinin tespit edilmesi, gözlem alanı hakkında daha ayrıntılı ve gerçekçi sonuçlar elde edilmesini sağlar.

Bu tez çalışmasının ikinci aşamasında gerçekleştirilen Otomatik Çizgi Tanıma yazılımı, şu anda XCLASS'ta herhangi bir şekilde başka bir yöntemle yapılamamaktadır. Son olarak, ALMA projesinin radyo teleskoplarından elde edilen toplam kütlesi güneşin kütlesinin 3 milyon katı olan ve galaksideki en karmaşık bölgelerden biri olan Sagittarius B2 (SgrB2) (Chown, 2012; Schilke, 2019) adlı büyük kütleli yıldız oluşum bölgesi üzerindeki verileri modelleyen XCLASS'ın üzerinde çalışıldığı için İstanbul Üniversitesi-Cerrahpaşa bu tez çalışması ile astroinformatik alanında uluslararası bir projeye dahil olmuştur. Bu kadar büyük ve karmaşık bir bölgeden elde edilen verilerin başarılı bir şekilde analiz edilmesi, evrenin farklı bölgelerinde de analiz yapılabileceği anlamına gelecektir. Böylece karanlık evrenin fiziksel ve kimyasal yapısı hakkında daha fazla bilgiyi daha kısa sürede elde etmek mümkün hale gelebilir.

Bunların yanı sıra, kurulumu tamamlanmakta olan Doğu Anadolu Gözlemevi'nin (DAG, 2019) 4m çaplı kızılötesi ve optik teleskobu ile yapılacak gözlemlerden elde edilecek verilerin incelenmesi aşamasında, bu proje bahse konu gözlemevine de faydalı olacak bilgi ve deneyim elde edilmesine de katkıda bulunabilir.

Sonraki alıřmalarda, 2007 yılında alıřmaların bařladıđı ve son 10 yıldır Avrupa’da ve Amerika’da alıřmaların ok hızlı bir řekilde devam ettiđi, paralel programlama tekniđinin son ve en st versiyonu olan kuantum hesaplama ile XCLASS’ın ve ok byk boyutlu verilerle alıřmak zorunda olan diđer tm yazılımların daha da hızlandırılmış versiyonlarının geliřtirilebilir. |



KAYNAKLAR

- [Academy, K., 2015, *Yıldızların ölçümleri-interferometry*, <http://www.khanacademy.org.tr/fen-mufredati/7.-sinif-fen-bilimleri/1.-unite:-gunes-sistemi-ve-otesi/gunes-sistemi-otesi:-gok-cisimleri/%C4%B0nterferometri-yildizlari-olcumleme/3025>, [Ziyaret tarihi: 4 Haziran 2019].
- Anıl, 2016, Spektrum Çizgileri Nedir? Ve Simsiyah olan spektrum Göstergeç Kâğıdı Ne Anlama Gelebilir?, <https://matkafasi.com/75876/spektrum-cizgeleri-simsiyah-spektrum-gostergec-gelebilir>, [Ziyaret Tarihi: 20.09.2020].
- Arimilli, R. K., Siegel, D. W., 2002, Symmetric Multiprocessing (SMP) System with Fully Interconnected Heterogenous Microprocessors, Patent Application Publication, 1(19).
- Bardeau, S., Pety J., Hily-Blant, P., ve Guilloteau, S., 2006, Class-Continuum and Line Analysis Single-dish Software, A gildas software, Version 1.1, <https://www.iram.fr/IRAMFR/GILDAS/doc/pdf/class.pdf>, [Ziyaret Tarihi: 20 Haziran 2020].
- Bardeau, S., Pety J., 2008, A quick introduction to GILDAS, <https://www.iram.fr/IRAMFR/GILDAS/doc/pdf/gildas-intro.pdf>, [Ziyaret Tarihi: 21 Haziran 2020]
- Bunch, B., H., Hellemans, A., 2004, *The History of Science and Technology*, Houghton Mifflin Harcourt, Boston, ISBN 978-0-618-22123-3.
- DAG, 2019, Doğu Anadolu Gözlemevi, <http://atasam.atauni.edu.tr/>, [Ziyaret Tarihi: 14.10.2020]
- Diaz, J., Munoz-Caro, C., Nino, A., 2012, A Survey of Parallel Programming Models and Tools in the Multi and Many-Core Era. *IEEE Transactions on Parallel and Distributed Systems*, 23(8), 1369–1386.
- Dickinson, M., Valdes, F., 2009, *A Guide to NEWFIRM Data Reduction with IRAF*, <http://ast.nao.edu/data/software>, [Ziyaret Tarihi: 21 Haziran 2020].
- Edgemont, R., 2020, *AIPS Cookbook*, The National Radio Astronomy Observatory, Charlottesville, VA 22903–2475.
- Ergün, U., Sayar, A., 2014, Fonksiyonel Programlama Dilleri ile Paralel Programlama, Niğde Üniversitesi Mühendislik Bilimleri Dergisi, 3(2), 1-17.
- Estalella, R., 2008, *Maximum angular size sensitivity of an interferometer*, <https://web.archive.org/web/20161014131518/http://www.am.ub.edu/~robert/Documents/umin.pdf>, [Ziyaret tarihi: 23 Mayıs 2019].
- George B. R., Lightman A. P., 1985, *Radiative Processes in Astrophysics*, WILEY-VCH Verlag GmbH & Co. KGaA , Germany, ISBN: 978-0-471-82759-7.
- Guilloteau, S., Lucas, R., Radford, S., 1999, An introduction to the IRAM Plateau de Bure interferometer, <ftp://ftp.eso.org/scisoft/scisoft4/sources/gildas/exe/doc/bure.ps>, [Ziyaret tarihi: 22 Eylül 2020].
- Gurwell, M. 2020, SMA observer Center, <http://sma1.sma.hawaii.edu/proposing.html>, [Ziyaret tarihi: 23 Eylül 2020].

- Hall, C., A., Meyer, W., W., 1976, Optimal Error Bounds for Cubic Spline Interpolation, *Journal of Approximation Theory*, 16(2): 105–122.
- Hutter, D., J., 2012, *Ground-based optical interferometry*, U. S. Naval Observatory Flagstaff Station, Scholarpedia, 7(6):10586.
- Jang, H., Park, A., Jung, K., 2008, Neural Network Implementation Using CUDA and OpenMP. 2008 Digital Image Computing: Techniques and Applications, 1-3 Aralık 2008 Canberra, ACT, Australia, ISBN: 978-0-7695-3456-5, IEEE, 155-161.
- Karaahmet, H.Y., 2017, Dalga Boyları, Frekans Bantları, <https://amatortelsizci.wordpress.com/2017/09/13/temel-teknik-bilgiler/>, [Ziyaret Tarihi: 4 Kasım 2020].
- Lindner, R.R., Vera-Ciro, C., Murray, C.E., Stanimirović, S., Babler, B., Heiles, C., Hennebelle, P., Goss, W.M., Dickey, J., 2015, Autonomous Gaussian Decomposition, *The Astronomical Journal*, 149:138.
- Möller, T., 2018, An automatic line identification function for XCLASS, I. Physikalisches Institut, Universität zu Köln.
- Möller, T., Endres, C., Schilke, P., 2015, eXtended CASA Line Analysis Software Suite (XCLASS), *Astronomy & Astrophysics*, A7, 598-616.
- Möller, T., Schilke, P., 2017, Manual for XCLASS-Interface Version 1.2.4, I. Physikalisches Institut, Universität zu Köln, Produced for ALMA.
- Mundy, L., 2013, Combined Array for Research in Millimeter-wave Astronomy, https://sites.nationalacademies.org/cs/groups/bpasite/documents/webpage/bpa_070721.pdf, [Ziyaret tarihi: 20 Eylül 2020].
- Mühle, S., Alves, F., Burkutean, S., Müller, H., Möller, T. 2013, ALMA Tools, Within the German Node of the European ALMA Regional Center Argelander-Institut für Astronomie der Universität Bonn, I. Physikalisches Institut der Universität zu Köln, BMBF-VF.
- NRAO, 2009, Very Large Array, <http://www.vla.nrao.edu/>, [Ziyaret tarihi: 23 Eylül 2020].
- Schieven, G., 2015, Observing with ALMA: A Primer for Early Science, ALMA Doc. 3.1, ver. 4.
- Sandersand J., Kandrot E., 2010, CUDA by Example: An Introduction to General Purpose GPU Programming, NVIDIA Corporation, Boston, ISBN: 978-0-13-138768-3.
- Sánchez-Monge Á., 2018, XCLASS: automatic line fitting of ALMA data Introduction and Tutorial, I. Physikalisches Institution Universität zu Köln.
- Schilke, P., 2011, Sagittarius B2 (SgrB2), <https://astro.uni-koeln.de/schilke/research/sgr-b2-a-super-star-cluster-in-the-making>, [Ziyaret Tarihi: 18.10.2020]
- RF Wireless World, 2017, Advantages and disadvantages L, S, C, X, Ku, K, Ka frequency bands <https://www.rfwirelessworld.com/Terminology/Advantages-and-Disadvantages-of-L-S-C-X-Ku-K-Ka-Frequency-Bands.html>. [Ziyaret Tarihi: 23.09.2020].
- Ruetsch, G., Fatica, M., 2014, CUDA Fortran for Scientists and Engineers, NVIDIA Corporation, Santa Clara, USA, ISBN: 978-0-12-416970-8.

- Tennyson, J., 2005, *Astronomical Spectroscopy* (2. Baskı), Imperial College Press., University College London, UK, ISBN-13: 978-1860945137.
- Unsöld A., Baschek B., 1991, *Classical Astronomy. The Solar System*, The New Cosmos, Chapter 2, Heidelberg Science Library. Springer, Berlin, Heidelberg, Almanya, ISBN:978-3-662-02683-0, s:3-73.
- Uslu, İ., 2019, Spektral Analiz, https://www.academia.edu/1748303/Spektral_analiz, [Ziyaret Tarihi: 23 Eylül 2020].
- Winters, J.M., König, S., Neri, R., Guilloteau, S., Lucas R., Krips, M., 2019, An Introduction to the IRAM NOEMA interferometer, <http://www.iram.fr/IRAMFR/GILDAS/doc/pdf/noema-intro.pdf>, [Ziyaret tarihi: 22 Eylül 2020].



ÖZGEÇMİŞ

Kişisel Bilgiler			
Uyruğu	☒ T.C.	☐ Diğer:	

Eğitim Bilgileri	
Lisans	
Üniversite	
Fakülte	
Bölümü	

Yüksek Lisans		
Üniversite		
Enstitü Adı		
Anabilim Dalı	Anabilim Dalı Adı	
Programı		

Doktora		
Üniversite		
Enstitü Adı		
Anabilim Dalı	Anabilim Dalı Adı	

Makale ve Bildiriler	