

**GAZİ ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
MATEMATİK EĞİTİMİ ANABİLİM DALI**

**BİR ÖĞRETİM MATERYALİ OLARAK
BİLGİSAYAR DESTEKLİ MATEMATİK YAZILIMLARININ
DEĞERLENDİRİLMESİ**

YÜKSEK LİSANS TEZİ

**Hazırlayan
İnci ÇİFTÇİ**

**Danışman
Yrd. Doç. Dr. Tolga GÜYER**

ANKARA-2006

İnci ÇİFÇİ'nin *Bir Öğretim Materyali Olarak Bilgisayar Destekli Matematik Yazılımlarının Değerlendirilmesi* başlıklı tezi 17/11/2006 tarihinde, jürimiz tarafından Matematik Eğitimi Anabilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Adı Soyadı

İmza

Başkan :	Prof.Dr. Ziya ARGÜN	
Üye :	Doç.Dr. Ahmet ARIKAN	
Üye :	Yrd. Doç. Dr. Tolga GÜYER (Tez Danışmanı).....	

Prof. Dr. Necdet HAYTA
Enstitü Müdürü

ÖNSÖZ

Matematiğin öğretimi ve sevdirmesi, özellikle ilk ve ortaöğretim seviyelerinde görev yapan öğretmenler için her zaman oldukça zor ve zahmetli bir iş olmuştur. Elde edilen teorik bilgi birikimi kullanılarak hazırlanan eğitim materyalleri, geniş kitlelere uygulanabilirlikleri ölçüsünde başarılı olmuşlardır. Ancak günümüzde, matematik eğitiminde kullanımı ve kullanımının yaygınlaştırılması çok kolay bir materyal vardır; bilgisayar. Dolayısıyla bu çalışmada, matematik eğitiminde geleceğin vazgeçilmez araçları olacağına inandığımız bilgisayar destekli matematik yazılımlarının öğretim materyali olarak değerlendirilmesi yapılmıştır. Ayrıca, bilgisayar destekli matematiğin en popüler ürünleri olan bilgisayar cebiri sistemleri tanıtılmış ve bu sistemlerin işleyiş mantığını açıklamak amacıyla sembolik hesaplama kuramı temel seviyede ele alınmıştır. Bilgisayar yazılımlarının matematik öğretiminde önemini vurgulayan çeşitli çalışmalar incelenmiş ve son olarak güncel Bilgisayar Cebiri Sistemlerinde geliştirilmiş Bilgisayar Destekli Matematik Öğretimi materyal örnekleri verilmiştir.

Bu tezin gerçekleştirilmesinde bana büyük destek sağlayan tez danışmanım Sayın Yrd.Doç.Dr. Tolga GÜYER'e, ilgi ve yardımlarından ötürü Sayın Prof. Dr. Ziya ARGÜN ve Sayın Doç. Dr. Ahmet ARIKAN'a, ayrıca manevi desteğinden dolayı eşim Hakan ÇİFTÇİ'ye içtenlikle teşekkür ederim.

ÖZET

BİR ÖĞRETİM MATERYALİ OLARAK BİLGİSAYAR DESTEKLİ MATEMATİK YAZILIMLARININ DEĞERLENDİRİLMESİ

ÇİFTÇİ, İnci
Yüksek Lisans, Matematik Eğitimi Anabilim Dalı
Tez Danışmanı: Yrd. Doç. Dr. Tolga GÜYER
Eylül – 2006

Bu çalışmada, matematik eğitimi alanında kullanımları oldukça yaygınlaşmış olan bilgisayar cebiri sistemleri ele alınmıştır. Günümüzde en çok tercih edilen başlıca bilgisayar cebiri sistemleri genel özellikleri ile tanıtılmış, bu sistemlerin temelinde yer alan sembolik hesaplama teorisi giriş seviyesinde anlatılmıştır. Bilgisayarların matematiğin öğretimindeki rollerini vurgulayan araştırmalar incelenmiş, bu çalışmalar ışığında, bilgisayar cebiri sistemlerinin matematik eğitimindeki kullanımları değerlendirilmiştir.

ABSTRACT**EVALUATION OF THE COMPUTER AIDED MATHEMATICS SOFTWARE AS
AN INSTRUCTION MATERIAL**

ÇİFTÇİ, İnci

M.Sc., Mathematics Education

Advisor: Asist. Prof. Dr. Tolga GÜYER

September – 2006

In this study, the computer algebra systems which have become widely used in the mathematics education are considered. The most popular computer algebra systems are introduced with their fundamental properties and the theory of symbolic computation which is located in the origin of these systems is presented in the introductory level. The investigations which emphasize the role of the computers in the instruction of the mathematics are studied and the usage of the computer algebra systems in the mathematics education is evaluated in the light of these studies.

İÇİNDEKİLER

	Sayfa
JÜRİ ÜYELERİNİN İMZA SAYFASI	i
ÖNSÖZ	ii
ÖZET	iii
ABSTRACT	iv
TABLolar VE ŞEKİLLER LİSTESİ	vii
BÖLÜM	
1. GİRİŞ	1
2. BİLGİSAYAR DESTEKLİ ÖĞRETİM	3
Öğretim Teknolojileri	3
Bilgisayar Destekli Öğretim	5
Bilgisayar Destekli Öğretimin Önemi	7
Eğitsel Ders Yazılımları	10
3. BİLGİSAYAR CEBİRİ SİSTEMLERİ VE SEMBOLİK HESAPLAMA	13
Bilgisayar Cebiri Nedir?	13
Genel Amaçlı Bilgisayar Cebiri Sistemleri	15
Mathematica	15
Maple	16
MathCad	19
Özel Amaçlı Bilgisayar Cebiri Sistemleri	20
Genel Amaçlı Bilgisayar Cebiri Sistemlerinin Bazı Ortak Özellikleri	21
Karar Verme	21
Grafik Çizimleri	22
Matematiksel Yazım Formatı	23
Uygulamalar	24
Program Kütüphaneleri	24
Karşılaşılan Sorunlar	24
Maple’da Temel Kavramlar	27
Sembolik Hesaplama Kuramının Temelleri	30

Sembolik Algoritmalar	34
4. MATEMATİK ÖĞRETİMİ VE BİLGİSAYAR	37
Bilgisayarın Kısa Bir Tarihçesi	37
Donanım Gelişimi	37
Yazılım Gelişimi	48
Bilgisayar Destekli Matematik Öğretimi	50
Bilgisayar Destekli Matematik Öğretimi Materyali Örnekleri	60
Maple’da Maplet Kavramı	61
Riemann Toplamları Mapleti	61
Adım Adım Öğreticiler	62
5. SONUÇ VE ÖNERİLER	67
KAYNAKLAR	69

TABLOLAR VE ŞEKİLLER LİSTESİ

	Sayfa
Tablo 1: Karşılaştırmalı sayısal ve sembolik hesaplama örnekleri	14
Şekil 1: Maple 10 dan genel görünüm	18
Şekil 2: Maple 10 da şekil tanıma özelliği	19
Şekil 3: Maple’da bir <i>tubeplot</i> örneği	23
Şekil 4: Matematiksel ifadelerin ikili ağaç gösterimi	30
Şekil 5: $x \sin(x^2)$ fonksiyonunun türevi için ikili ağaç gösterimi	33
Şekil 6: Leibniz çarkı	38
Şekil 7: Babbage’ın fark makinesi	39
Şekil 8: Mark-Ic	41
Şekil 9: ENIAC’dan çeşitli görüntüler	42
Şekil 10: EDVAC	43
Şekil 11: (a) UNIVAC,	
(b) UNIVAC sistemlerinde kullanılan gelişmiş bellek teknolojileri	44
Şekil 12: Apple-I bilgisayarı	46
Şekil 13: 1981 yılının Ağustos ayında piyasaya sürülen ilk IBM PC bilgisayarı .	47
Şekil 14: 80286 işlemcili IBM PC/AT	48
Şekil 15: ENIAC’da program yazan iki mühendis	49
Şekil 16: $f(x)$ ve $f(x)+2$ parabolleri	53
Şekil 17: Maple 10 da çizilmiş bir küre	54
Şekil 18: Crowe ve Zand’in sınıflandırması	55
Şekil 19: Excel’de fonksiyon grafikleri	57
Şekil 20: Excel’de fonksiyon grafikleri	58
Şekil 21: Riemann toplamaları mapleti	62
Şekil 22: Adım adım “Limit Hesaplama” öğreticisi	64
Şekil 23: Adım adım “Türev Alma” öğreticisi	65
Şekil 24: Adım adım “İntegral Alma” öğreticisi	66

BÖLÜM 1

GİRİŞ

Pozitif bilimlerin en eskilerinden birisi olan matematik, soyut doğası gereği her zaman öğrenilmesi ve öğretilmesi zor olan alanlardan birisi olmuştur. Bununla birlikte, matematik eğitimcileri bilgileri öğrencilerine aktarabilmek, özellikle de soyut kavramları onların akıllarında somutlaştırabilmek için diğer alanlarda olduğu kadar eğitim materyaline sahip olamamışlardır. Kuşkusuz ki matematiğin, her seviyedeki öğrenen tarafından, “anlaşılması zor”, “karmaşık” ve “sıkıcı” gibi sıfatlarla nitelendirilmesinin ardındaki en önemli sebeplerden birisi de, onu gerçekten anlayan ve öğretme kaygısı içinde olan çok az sayıda insanın, matematiği öğretme konusunda yaşadıkları bu zorluklardır.

Bu konuda yakın bir geçmişe sahip en önemli atılımlardan birisi de matematik öğretmenlerini (ve temel seviyede matematik anlatan sınıf öğretmenlerini), anlatmak istedikleri konuyu daha iyi öğretebilmelerini sağlayacak çeşitli materyaller hazırlayabilecek doğrultuda eğitmek olmuştur. Matematik öğretmenlerine “doğrudan üretme” şeklinde yeni bir sorumluluk veren bu uygulamanın başarısını olumsuz yönde etkileyen temel sebeplerden birisi, uygulamanın yeni mezun genç öğretmenlerin önemli bir kısmı tarafından benimsenmesine karşın, tecrübelerine dayanarak kendi öğretme yöntemlerini geliştirmiş eski öğretmenler tarafından çok da benimsenmemiş olmasıdır.

Bu bağlamda bilgisayarlar, geleceğin matematik eğitimcileri için en vazgeçilmez eğitim materyali olma konusunda hızla ilerleme kaydetmektedirler. 70’li yıllarda başlayan ve son on yıl içerisinde oldukça gelişme gösteren “matematik yazılımları”, şüphesiz ki bu ilerlemenin en önemli sebeplerinden birisidir. Ancak

burada kastedilen yazılımlar, daha çok nümerik analiz tabanına oturan ve sayısal hesaplamalarda kullanılan, girdi ve çıktı verileri tümüyle ondalık sayılardan oluşan yazılımlar değildir. Bu çalışmalara paralel olarak yürütülen ve günümüzde uygulamalı matematiğin yeni bir disiplini haline gelen “sembolik hesaplama” ya da “bilgisayar cebiri” teorileri, yazılım teknolojilerindeki gelişmelere bağlı olarak gerçekten “işe yarar” ürünlerini çok yakın tarihlerde çıkarmaya başlamışlardır.

Bilgisayar cebiri yazılımlarındaki bu gelişmeler, iki önemli konu üzerinde olumlu etkilere yol açmıştır. Bu yazılımların matematiğin uygulamalı alanlarında kullanım oranları artmış, dolayısıyla da daha fazla matematikçinin güvenini kazanmışlardır. Diğer yandan bu yazılımlar, son yıllarda matematik eğitimcilerinin de dikkatini çekmeyi başarmışlar, bu alanda yapılan çalışmalarda sıklıkla Maple, Mathematica, Mathcad gibi popüler bilgisayar cebiri yazılımlarının adları geçmeye başlamıştır. Amacı, bilgisayar cebiri sistemlerinin temelleri hakkında genel bir bilgi verip, bu konunun eğitim yönüyle ele alındığı yurtdışı kaynaklı çalışmaları değerlendiren ve ülkemizde gerçekleştirilebilecek uygulamalar hakkında öneriler sunan bu tez, beş bölümden oluşmaktadır:

Birinci bölüm giriştir. İkinci bölümde genel olarak bilgisayar destekli eğitim literatürü verilmiştir. Üçüncü bölümde “bilgisayar cebiri” ve “sembolik hesaplama” kavramları anlatılmış, popüler bilgisayar cebiri sistemleri tanıtılmıştır. Dördüncü bölüm, bilgisayar destekli matematik eğitimi üzerine yapılan çalışmaların değerlendirilmesine ayrılmıştır. Son bölüm olan beşinci bölümde ise, sonuç ve öneriler verilmiştir.

BÖLÜM 2

BİLGİSAYAR DESTEKLİ ÖĞRETİM

2.1. Öğretim Teknolojileri

21. yüzyılın başında olduğumuz şu günlerde dünyada hem kültürel, hem teknolojik hem de eğitim alanında büyük gelişmeler yaşanmaktadır. Değişime ayak uyduran ülkeler katlanarak güçlenmekte, değişimi yakalayamayan ülkeler ise çağın gerisinde kalmaktadır. Gelişmekte olan ülkelerin ileriye dönük pek çok projeleri eğitim konusunda yoğunlaşmaktadır. Ülkemizin de 2000’li yıllara ait amaçlarından birisi, değişen ve gelişen dünya şartlarına uyum sağlayabilecek, temeli bu amaç üzerine kurulmuş, çağdaş ve bilimsel bir eğitim sisteminin oluşturulmasıdır. Öğrencilerimizin bağımsız düşünen ve düşündüklerini kolaylıkla dile getirebilen, eleştiri yapabilen, kendisine güvenen, üretken, lider, problemlerini kendi başına çözebilen ve sosyal kişilikte olabilmeleri için, çağdaş eğitim sistemlerine ihtiyaç vardır.

Teknolojinin “bilimsel araştırmalarla elde edilen sistematik bilgilerin pratik alanlara uygulanması” (Heinich vd., 1993) tanımını kabul edersek, öğretim teknolojisinin öğretimde kullanılan makineler ya da materyaller olarak tanımlanmasının doğru bir yaklaşım olamayacağı ortaya çıkar. Çünkü bütün bu araçlar ya da materyallerin kendi teknolojileri, nasıl tasarlandığı ve oluşturulduğuna yönelik sistematik bilgileri vardır. Dolayısıyla öğretim teknolojisi, öğrenen üzerinde, özellikle öğrenme sonucu kazanılan yeterlikler ve davranışlar üzerinde odaklanmalıdır. Bu alanın sistematik bilgileri, insan davranışlarındaki öğrenme sonucunda oluşan değişmeyi nasıl meydana getireceğimizi açıklayan davranış

bilimleri ile ilgili bilimsel arařtırmalardan elde edilen teknik ve süreçlerin bütününden meydana gelmelidir (Yalın, 2001).

Klasik öğretim biçiminde sınıf eğitimi öğretmen merkezli yapılmaktadır. Öğretmen konunun tek hâkimi ve konuyu aktaran tek kaynak olarak görölmektedir. Öğretmenin gereğinden fazla sorumluluğunun bulunması, öğrencinin kendi çabaları ile araştırma yapmasına engel olmaktadır. Klasik eğitim sisteminde planlayıcı, eğitici, danışman ve yönetici konumunda olan öğretmenin rolü değişmeye başlamıştır. Öğretmen bilgiyi öğrenciye aktaran tek kaynak olmaktan çıkmış, öğrenciyi bilgiye yönlendiren kişi konumuna gelmiştir. Öğrenci ise, dinleyici durumunda pasif olmaktan kurtularak; bilgiye ulaşması gereken ve konu hakkında alternatif düşünceler üreten kişi halini almıştır. Öğrenme ve öğretme kavramının yorumlanmasında meydana gelen bu değişiklikler öğretim teknolojilerinin ortaya çıkmasına ve gelişmesine zemin hazırlamıştır (Carr,1996).

Günümüzde çağdaş eğitim anlayışının bir gereği olarak öğretim hizmetlerinin düzenlenmesinde öğretim teknolojilerinden yararlanılması eğitim programlarının vazgeçilmez bir parçası olarak görölmektedir. Bilim ve teknolojiadaki gelişmeler eğitim ve öğretim ortamını doğrudan etkilediği için, son dönemde modern yaşamın ayrılmaz bir parçası olan bilgisayar, eğitim ve öğretim ortamında da kendisinden söz ettiren bir araç haline gelmiştir.

Diğer yandan, “öğretim teknolojisi” ve “eğitim teknolojisi” kavramları arasında kesin ve net bir ayırt konusunda görüş birliği oluşmadığından, bu iki kavram genellikle birbirleri yerine kullanılmaktadır. Alkan (1995, s.19), bu kavramların aslında birbirlerinden farklı olduklarını savunarak, bu farkları şu şekilde açıklamaktadır:

“Öğretim teknolojisi”, öğretimin, eğitimin bir alt kavramı olduğu anlayışına dayalı olarak ve belirli öğretim disiplinlerinin kendine özgü yönlerini dikkate alarak düzenlenmiş teknolojiyle ilgili bir terimdir. Örneğin “fen öğretimi teknolojisi”, “dil öğretimi teknolojisi”, “biyoloji öğretimi teknolojisi” gibi. Bu terim, ilgili disiplin alanlarına özgü olarak etkili

öğrenme düzenlemeleri oluşturmak üzere amaçlı ve kontrollü durumlarda insan gücü ve insan gücü dışındaki kaynakları birlikte işe koşarak belirli özel hedefler doğrultusunda öğrenme-öğretme süreçleri tasarlama, işe koşma, değerlendirme ve geliştirme eylemlerinin bütününe içeren sistematik bir yaklaşımı ifade etmektedir.

“Eğitim teknolojisi” ise “insanın öğrenmesi” olgusunun tüm yönlerini içeren problemleri sistematik olarak analiz etmek, bunlara çözümler geliştirmek üzere ilgili tüm unsurları (insan gücünü, bilgileri, yöntemleri, teknikleri, araç-gereçleri, düzenlemeleri vb.) işe koşarak uygun tasarımlar geliştiren, uygulayan, değerlendiren ve yöneten karmaşık bir süreçtir. Diğer bir deyişle “eğitim teknolojisi” terimi, öğretme-öğrenme süreçleri ile ilgili özgün bir disiplini vurgularken, “öğretim teknolojisi” terimi ise bir konunun öğretimi ile ilgili öğrenmenin kılavuzlanması etkinliğini ifade etmektedir.

Bugün eğitim ve öğretim teknolojisi kavramları akla ilk olarak bilgisayarları getirmektedir. Eğitimde bilgisayar kullanımı bilgiye ulaşım ve bilgilerin iletimi açısından büyük kolaylıklar sağlayacaktır. Eğitime olan talebin ve öğrenci sayısının hızla artması, bilgi miktarının çoğalması ve içeriğinin karmaşık hale gelmesi, bireysel yeteneklerin ve farklılıkların giderek daha çok önem kazanması gibi nedenler, eğitimde bilgisayar kullanımını zorunlu hale getirmiştir.

20. yüzyılın ortalarından itibaren eğitimde verimliliği artırmak amacına yönelik birçok araştırma yapılmıştır. Öğrenim süreci içerisinde öğrencinin daha aktif olmasını sağlamak ve öğrenmeyi daha etkin hale getirmek amacıyla farklı öğretim stratejileri geliştirilmiştir. Bu stratejilerin en önemlilerinden bir tanesi de bilgisayar destekli öğretimdir (Fidan,1983, s.13).

2.2. Bilgisayar Destekli Öğretim

Hızla gelişen bilgi çağına ayak uydurabilmek için bilimsel ve çağdaş eğitim sistemlerine ve öğretim teknolojilerine ihtiyaç duyulmaktadır. Teknoloji destekli ve teknoloji tabanlı öğretim ortamları hazırlayarak eğitimin kalitesinin artırmak bulunduğumuz yüzyılın toplumları için zorunlu görünmektedir. İdeal bilgi toplumu

oluşturabilmek için gerekli personelin yetiştirilmesi, eğitim ortamlarının modernleşmesine bağlıdır. Bu amaçla kullanılabilecek eğitim teknolojilerinden birisi de bilgisayarlardır. Bilginin üretilmesinde ve iletilmesinde önemli payı olan elektronik alanda, son yıllarda görülen buluşlar, kaydedilen gelişmeler sonucunda, gelişmiş ya da gelişmekte olan tüm ülkeler hızlı bilgisayarlaşma sürecine girmiştir (Akkoyunlu,1995).

Bilgisayarların öğretimde öğretmene yardımcı olarak kullanılmasına “bilgisayar destekli öğretim” denir (Erden,1994). Bilgisayarlar bir konu ile ilgili olgu, kavram, ilke, genelleme ve konuların öğretilmesinde bire-bir öğretim amacıyla kullanılabildiği gibi işlenmiş konularla ilgili alıştırmaya ve tekrarlarla da kullanılmaktadır. Öğrenme ortamında alıştırmaya yapmak, problem çözmeye yönelik üst düzey davranışların kazandırılmasına yardımcı olmakta ve öğrenmede kalıcılığı sağlamaktadır.

Bilgisayar ortamı söz konusu olduğunda, donanım ve yazılım kavramlarından söz etmek zorundayız. Donanım ve yazılım ikilisi bilgilerin sunumu, kontrolü, depolanması ve bireylerin yanıtlarını kontrol etmek için tasarlanmış aygıtlar ve platformlardır. İşlem yapabilir ve dönüt verebilirler. Teknolojideki hızlı ilerleme sonucunda donanımdaki nitelik ve kapasite de sürekli olarak gelişmektedir.1950’lerde ENIAC gibi tonlarca ağırlığı olan makineler bilgisayar donanımı olarak kullanılıyorken, günümüzde ceplere sığabilen, daha hızlı, işlevsel ve ekonomik donanımlar bulunmaktadır. Bugün gelişmiş ve yeterli sayılan bir donanım elemanı, birkaç yıl içinde yetersiz ve demode duruma düşmektedir. Dolayısıyla bilgisayar destekli eğitim ile ilgili donanım seçerken standart ve özellikleri başlangıçta çok dikkatli belirlemek gerekmektedir. Donanımın temel elemanlarının uzun süre kullanılabilmesi ve kolayca genişletilebilecek yapıda olmasına dikkat edilmelidir (Akpınar, 1995, s.16).

İşlevsel bir bilgisayarın aşağıdaki ana parçaları olmak zorundadır:

- Bilginin depolanacağı bir depo (disk, disket,CD-ROM gibi)

- Bilgilerin sunulacağı ya da gösterileceği bir ekran
- Yanıtların veya iletilerin girilebileceği bir girdi kabul sistemi (klavye, fare, mikrofon gibi)
- İşlemleri yapan bir işlemci(yongalar, devreler, değişik bellekler gibi)

Ancak bu cihazların en önemli parçası yazılımdır. Yazılım olmadan donanım hiçbir işe yaramamaktadır. İşleyen bir bilgisayar sistemi donanım ve yazılımı kapsar. Kullanıcı ve bilgisayar arasındaki iletişimin ana unsuru olan yazılım, donanımın işlevini yerine getirmesinde yol gösteren ve program adı verilen bir komut dizgesidir.

Eğitsel bir yazılım ise, öğretilecek olan konu alanıyla ilgili bilgi örüntülerini, öğrenci- donanım etkileşiminin içeriğini ve sürecini içeren program kodları setidir (Akpınar, 1995).

2.3. Bilgisayar Destekli Öğretimin Önemi

Bilgisayar Destekli Öğretim, bilgi teknolojileri çağını yakalayacak ve geçecek insan gücünün yetiştirilmesinde üzerinde durulan önemli bir konudur. Bilgisayar destekli öğretimin öğrenciler için hedeflenen genel amaçları şunlardır:

- Öğrencide ileri düzeyde düşünme becerisinin geliştirilmesini desteklemek
- Grup çalışmalarını desteklemek
- Mantık yoluyla problemlere çözüm bulmayı desteklemek
- Öğrencinin motivasyonunu artırmak
- Hipotez kurmaya cesaretlendirmek
- Öğrencinin bilimsel düşünme yeteneğini geliştirmek
- Öğretme yöntemlerini genişletmek
- Öğrencinin kendi kendine öğrenme yeteneklerini geliştirmek

Bilgisayar, öğrenme ve öğretme açısından diğer öğretim araçlarından farklı olarak benzersiz olanaklar sunan çok yönlü bir araçtır. Bilgisayarı diğer araçlardan ayıran en önemli özeliği bir üretim, öğretim, yönetim, sunu ve iletişim aracı olarak kullanılabilmesidir. Öğrenci merkezli eğitim sistemlerinin temeli, bireysel gereksinimlerin dikkate alınarak, öğrencinin kendine uygun biçimde ve hızda öğrenmesidir.

Bilgisayar teknolojisinin daha ucuzlayarak zengin olanaklarla bireylerin kullanımına sunulduğu, bilişim olanaklarının hızla gelişip yaygınlaştığı ve özellikle eğitsel yazılımların gün geçtikçe öğrenciyi daha çok dikkate aldığı bir çağda bulunuyoruz. Dolayısıyla öğrencinin kapasitesine, bireysel gereksinimlerine yanıt verebilen yazılımlar öğretmene rolünü değiştirmede yardımcı olabilir. Zeki yazılımlar sanal ortam ve çoklu ortam olanaklarını kullanarak öğretmenin rutin iş yükünü azaltacaktır.

Genel olarak Bilgisayar destekli öğretimin önemi aşağıdaki gibi sıralanmıştır:

- Hem bireysel öğretimi hem de grup öğretimini birlikte gerçekleştirmeye olanak sağlar. Öğrencinin hangi konuya ne kadar zaman harcayacağını, hangi alıştırma ne kadar tekrar edeceğini kendi çalışma ve öğrenme hızına göre ayarlayabilmesi bilgisayarın sunduğu zaman esnekliği sayesinde mümkündür. Öğrencinin bir gruba bağlı kalmadan bireysel öğrenme hızı ve yeteneğine göre ilerlemesini sağlarken “bireysel öğretim”i, öğretmeni ve grup arkadaşları ile birlikte çalışabilme olanağı sağlarken de “grup öğretimi”ni gerçekleştirme olanağını sunmaktadır.

- Bilgisayar destekli öğretim, küçük adımlar ilkesi ile tam öğrenmenin gerçekleşmesine olanak sağlar.

Öğretilecek konular basitten karmaşığa doğru en küçük birime ayrılarak sunulduğu ve her birim sonunda öğrenciye soru sorularak konuyu öğrenip öğrenmediği sınındığı için öğrenci bir sonraki konu birimine ancak bir önceki konu

birimini tam olarak öğrendikten sonra geçebilmektedir. Bu sistem, öğrenciye birim içinde anlamadığı noktayı istediği kadar tekrar edebilme olanağını sunar.

- Anında dönüt, düzeltme ve pekiştireç sunarak öğrenmenin tam olarak gerçekleşmesini sağlar. En iyi öğretmen bile öğrencilerin sınav veya çalışmalarını ancak bir süre sonra değerlendirip geri verebilirken, bilgisayar öğrenciye başarı durumu ile ilgili dönütü hemen vererek öğrencinin yaptığı yanlışın anında düzeltilmesini sağlar. Ayrıca Bilgisayar destekli öğretim’de öğrencinin verdiği yanıtlara anında pekiştireç verilerek öğrencinin başarı duygusu tatmin edilir.

- Öğretim ortamının zenginleştirilmesini sağlar. Bilgisayar kapasitesi açısından oldukça fazla sayıda etkinlik sunabilme özelliğine sahip olması nedeniyle, öğretim ortamında kullanılan diğer araç ve gereçlerden farklı bir eğitim aracıdır ki, bu da öğretim ortamının zenginleştirilmesini sağlar.

- Bilgisayarların tepki hızı yüksektir ve bu nedenle öğrenme hızlanır. Bilgisayarlar doğru ve hızlı geri bildirimler vererek, öğrencilerin doğru ve kısa sürede öğrenmelerini sağlarlar. Eğitim psikolojisi bulgularına göre bir yanlış yapıldığı anda ortaya çıkarmak ve doğrusunu göstermek, yanlış önlemenin en iyi yoludur. Bilgisayarda öğrenci çok hızlı bir geri besleme alır. Öğrenciler bir konuda yanlış bir iş yaptıklarında bilgisayar anında mesaj vererek doğruyu bulma yönünde yol gösterici ve uyarıcı bir rol oynar.

- Bilgisayarlar daima kullanıma hazır durumdadırlar; yorulmazlar, sıkılmazlar, dinlenmek için araya ihtiyaç duymazlar.

- Bilgisayarlar sayesinde öğrenciler deneyler yaparak neden- sonuç ilişkilerini görebilirler. Normalde dünyada yapılması zor ya da sınıf ortamında yapılması imkânsız olan deneyler bilgisayarlar ile çok ucuza mal edilerek ve zaman kaybı olmadan yapılabilir.

2.4. Eğitsel Ders Yazılımları

Bilgisayar destekli öğretimin öğrenci başarısı açısından geleneksel eğitimden daha etkili olduğu pek çok araştırmacı tarafından vurgulanmaktadır. Bilgisayar destekli öğretim ayrıca öğretimde zaman kazandırma ve öğrencide öğrenmeye karşı olumlu davranışlar ortaya koyması açısından da öğretime katkılar sağlamaktadır.

Kuşkusuz ki bilgisayar destekli öğretimin en önemli araçlarında birisi de eğitsel ders yazılımlarıdır. Son yıllarda kullanımları oldukça yaygınlaşan eğitsel ders yazılımları, eğitim teorilerine ve öğretim tasarımı ilkelerine uygun hazırlandıkları takdirde oldukça faydalı olmaktadır. Bu materyallerden, örgün eğitimin yerini almalarının beklenemeyeceği açıktır, ancak “öğretime destek” olarak, öğretim teknolojilerinin önemli bir ayağını oluştururlar.

Bilgisayar destekli öğretimde en çok kullanılan eğitsel ders yazılım türlerini aşağıdaki gibi sıralayabiliriz (Alessi ve Trollip, 1991):

a) Özel Eğitsel Ders Yazılımları: Bilgisayar Destekli Öğretimde en çok kullanılan yazılım türüdür ve belirli bir konu ya da kavramı öğretmeye yönelik programlardır. Özel ders yazılımları ders hakkında genel bilgi veren ve öğrencinin dikkatini çeken bir giriş bölümü ile başlar, bundan sonraki akış içerisinde her adımda öğrenciye bilgi sunma, bu bilgiye yönelik soru sorma, öğrencinin cevabını alma, cevabı değerlendirme ve uygun bir geri bildirim verme etkinlikleri yer alır. Bu döngü, program ya da öğrenci tarafından dersin bitirilmesine dek devam eder.

b) Alıştırma Yazılımları: Alıştırmalar, belirli bir konu ya da kavramı öğretmek yerine önceden sınıf ya da başka bir öğretim ortamında öğretilen konu ya da kavramı pekiştirmek amacıyla geliştirilen programlardır. Alıştırmalar genellikle tanımlar, tarihi olgular, matematik problemlerinin çözümü, bilimsel ilke ve kavramlar, dil öğretimi gibi alanlarda kullanılır. Alıştırma ders yazılımları, özel eğitsel ders yazılımlarında olduğu gibi öğrencinin dikkatini çeken, derse karşı ilgi uyandıran ve dersin amaçları hakkında genel bilgi veren bir giriş bölümü ile başlar.

Bundan sonraki genel akış içinde, her bir adımda, öğrenciye belirli bir konu ile ilgili soru sorma, öğrencinin cevabını doğru ya da yanlış olarak değerlendirme ve uygun geri bildirim verme etkinlikleri yer alır.

c) Benzetişim (Simülasyon) Yazılımları: Bilgisayar destekli öğretim’de benzetişimler, bir takım olay ve durumları modelleyerek öğrenciye bu olay ve durumlar hakkında bilgi ve beceri kazandırmayı amaçlayan eğitsel ders yazılımlarıdır. Benzetişim yazılımlarında öğrenciler karmaşık becerileri gerçek durumlarla karşı karşıya gelerek öğrenmekte, bir oyun çerçevesinde belli roller olarak sosyal, ekonomik ve çevre sorunlarını önlemeye çalışmaktadırlar. Benzetişim yazılımları öğrencilerin, konuların değişik boyutlarını görmesini sağlamakta, öğrenilenlerin genellenmesini kolaylaştırmaktadır.

Bir çalışmada Özdener (2005), benzetişim yazılımlarının önemini aşağıdaki cümlelerle vurgulamıştır:

“Bir diğer araştırma bulgusuna göre benzetişim yazılımı kullanan deney grubu öğrencileri ile gösteri yöntemi kullanılan kontrol grubu öğrencileri arasında, sınav genel başarısı açısından deney grubu lehine anlamlı fark görülmüştür. Öğrencilerle yapılan bireysel görüşme ve gözlemler bu tür eğitsel materyallerin motivasyona katkı sağladığı doğrultusundadır. Elde edilen bulgular ışığında; hazırlanan benzetişim yazılımının, öğretici özelliğe sahip olup, eğitime katkı sağladığı ve motivasyonu artırdığı söylenebilir. Araştırma sonuçları, benzetişim yardımı ile geliştirilen sanal laboratuvarların malzeme ve öğrenci performansı açısından bakıldığında birçok avantaja sahip olduğunu tespit eden literatürdeki pek çok araştırmayı destekler niteliktedir (Joseph, 1999; Özdener ve Erdoğan, 2001).

Okullarda bulunan laboratuvar eksikliği, malzeme yetersizliği ve kalabalık sınıflardan kaynaklanan nedenlerle, deneylerin ancak gösteri yöntemi ile gerçekleştirilebildiği düşünüldüğünde, benzetişim yazılımlarının gösteri yöntemine alternatif olabileceği açıktır. Ayrıca yüksek maliyetli laboratuvar araç-gereçleri nedeniyle sanal ortamların kullanılması, maliyetler bakımından da avantaj sağlayacaktır. Tüm bu yönleriyle, hazırlanan benzetişim yazılımı öğretim sürecinde kullanılabilir niteliktedir.”

Eğitsel ders yazılımlarının yanı sıra, son yıllarda bilgisayar dünyasında gerçekleşen teknolojik ilerlemelere paralel olarak gelişen “sanal gerçeklik” (virtual reality) kavramı da eğitimde kullanılmaya başlanmıştır. Oldukça olumlu sonuçlar alınan bu çalışmalar hakkında Kayabaşı (2005), bir makalesinde sanal gerçekliğin eğitimde kullanılmasının geleceği hakkında aşağıdaki yorumları yapmaktadır:

“Sanal gerçeklik, pahalı bir teknoloji olmasına karşılık birçok alanda eğitim-öğretim amaçlı olarak sınırlı da olsa kullanılmaktadır. Yani geleceğin yeni öğretim ortamlarını oluşturacağına hiç kuşku yoktur.

Sanal gerçeklik, örneğin bir yapı daha inşa edilmeden, ekranda olan bir model binanın içine girerek gezmek, fonksiyonel olup olmadığını incelemek, günün değişik saatlerinde güneşin etkisinin nasıl olduğunu bilmek veya binanın belirli derecedeki depreme dayanımını ölçmek için tasarım özelliklerini kritik edebileceğimiz bir ortamı sağlar. Ayrıca renk, aydınlatma ve ergonomi gibi faktörlerin de önceden denenmesi mümkündür.

Okullarda bu sistem kullanıldığı takdirde, öğrencilerin motivasyonu artırılarak öğrenmelerde kalıcılık sağlanmış olacaktır. Sanal gerçeklik yeni bir alan ve yeni bir teknolojik gelişme olmasına rağmen öğretim materyali olarak, yararlarının ölçülmesi için bir takım çalışmalar halen devam etmektedir.

Sanal gerçeklik, kullanıcının, görsel, duygusal, dokunmayla, koku ve tatla algılanan hissel seçeneklerle etkileşimde bulunarak, gerçek zaman benzetişimleriyle ilişkilendirilmesidir. Sanal gerçeklik bunların yanında, askeri alanlarda sanal olarak senaryolar doğrultusunda oluşturulmuş savaşa hazırlık çalışmalarında, mühendislik alanlarında özellikle nükleer çalışmalarda, tıp eğitiminde özellikle uygulamalı tıpta yaygın olarak kullanılmaktadır. Öğrenciler sanal ortamlarda oluşturulmuş kadavralarda istedikleri çalışmaları yaparlar ya da insan vücuduna bir yolculuk yapabilirler. Yine yabancı dil eğitiminde öğrencilerin eğitilmesi amacıyla etkin bir şekilde kullanılabilir. Öyle ki, yabancı dil öğrenmek isteyen bir öğrenciye, sanal eğitim yoluyla dilini öğrenmek istediği ülkeye yolculuk yaptırılarak, o ülkede yaşama fırsatı verilebilir. Yine bir doktor çok tehlikeli bir ameliyatı kolayca, tehlikesizce ve tekrar tekrar yapabilir.”

BÖLÜM 3

BİLGİSAYAR CEBİRİ SİSTEMLERİ VE SEMBOLİK HESAPLAMA

3.1. Bilgisayar Cebiri Nedir?

Sayısal yöntemlerde kullanılan hesaplamalar, temel aritmetik işlemlerin yanı sıra matematiksel fonksiyonların sayısal değerlerinin hesaplanması, polinomların köklerinin bulunması, başlangıç-değer problemlerinin çözülmesi, sayısal integrasyon ve matrislerin sayısal öz değerlerinin hesaplanması gibi daha karmaşık işlemleri de içerirler. Ancak bütün bu işlemlerin ortak bir noktası vardır; *sayılar*. Hesaplamalar sadece sayılar üzerinde gerçekleştirilmektedir. Ayrıca bu hesaplamalar çoğunlukla “kesin” değildirler. Çünkü veriler çoğunlukla kayan-noktalı (floating-point) sayılar içerirler ve yapılan işlemler, adım sayısı arttıkça aynı oranda büyüyen bir hata payını da beraberlerinde getireceklerdir. Ancak hata analizleri sonucu elde edilen hata kestirme formülleri sayesinde, hatanın kabul edilebilir sınırlar içerisinde tutulması mümkün olmaktadır.

Matematiksel hesaplamanın diğer bir araştırma alanı, “Sembolik ve Cebirsel Hesaplama” ya da “Bilgisayar Cebiri” olarak adlandırılan ve kısaca, “matematiksel nesnelerin gösteriminde kullanılan semboller üzerinde işlem yapma” şeklinde tanımlanan yöntemleri içerir. Bu semboller tamsayılar, rasyonel sayılar, gerçel sayılar ya da karmaşık sayılar gibi sayıları gösteren semboller olabilecekleri gibi, polinomlar, rasyonel fonksiyonlar, denklem sistemleri gibi matematiksel nesneleri ya da gruplar, halkalar, cisimler gibi çok daha soyut cebirsel nesneleri gösteren semboller olabilirler (Davenport vd., 1993).

Sembolik kelimesi matematiksel problem çözmede ulaşılmak istenen son noktanın çoğu zaman kapalı ve simgesel bir formül biçiminde olduğunu vurgulamaktadır. Diğer bir deyişle ulaşılmak istenen sonuç, analitik olarak ifade edilebilmelidir. *Cebirsel* kelimesiyle ise hesaplamaların kayan-nokta aritmetiği yerine kesin sonuç adımları üzerine kurulu olduğu kastedilmektedir.

Örneğin $\sqrt{2}$ sembolü ondalık kısmı sonsuza kadar uzayıp giden 1,4142135623730... irrasyonel sayısını göstermektedir. Ancak bu sayısal değerini hiç kullanmadan bu sayıyı 2 ile çarpabilir, dolayısıyla yine bir irrasyonel sayı olan 2,82842712474619... sayısını gösteren yeni bir sembol, $2\sqrt{2}$ elde edilebilir. Görüldüğü gibi burada sayısal değerleri hiç kullanmadan doğrudan **sembollerini** kullanarak bir **hesaplama** gerçekleştirdik.

Tablo1’de, karşılaştırmalı olarak sayısal ve sembolik metotlar kullanılarak gerçekleştirilmiş bazı hesaplama örnekleri verilmiştir:

Tablo 1
Karşılaştırmalı Sayısal ve Sembolik Hesaplama Örnekleri

Sayısal	Sembolik
$2/6 \rightarrow 0.333333$	$2/6 \rightarrow 1/3$
$x+2x \rightarrow x=?$	$x+2x \rightarrow 3x$
$\text{Cos}(3.14159) \rightarrow -0.999999$	$\cos(\pi) \rightarrow -1$
$\int_0^{1/2} \frac{x}{x^2-1} dx \rightarrow 0.1438$	$\int \frac{x}{x^2-1} dx \rightarrow \frac{\ln x^2-1 }{2} + c$
$\left. \frac{dx^2}{dx} \right _{x=2} \rightarrow 4$	$\frac{dx^2}{dx} \rightarrow 2x$

Sembolik hesaplamada karşılaşılan en büyük güçlük, semboller üzerinde hesaplama ve karar verme gibi işlemleri gerçekleştirecek algoritmaların

tasarlanmasıdır. Bu algoritmalar için girdi verileri, programlama dillerinde *string* (karakter dizileri) olarak tabir edilen türdeki verilerdir. Dolayısıyla bu verilerin sembolik algoritmaya verilebilecek “bilgi” biçimine dönüştürülebilmesi için, öncelikle anlamlı bir matematiksel yapı haline getirilmesi gerekir. Daha ileri kesimlerde anlatılacak olan bu yapılar, matematiksel ifadelerin ikili ağaç gösterimleri olacaktır.

3.2. Genel Amaçlı Bilgisayar Cebiri Sistemleri

Amaçları sembolik hesaplama işlemlerini gerçekleştirmek olan, ancak bunun yanı sıra sayısal hesaplamaları da yapabilen bilgisayar yazılımları genel olarak bilgisayar cebiri sistemleri olarak adlandırılırlar (Davenport vd., 1993). Günümüzde yaygın olarak kullanılan başlıca genel amaçlı bilgisayar cebiri sistemleri:

- Maple
- Mathematica
- Matlab
- MathCad
- Reduce
- Macsyma
- Magma
- Axiom

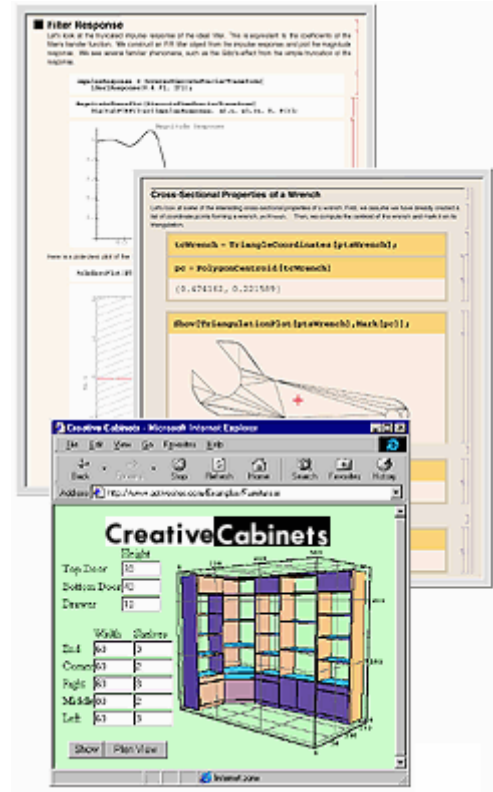
olarak sıralanabilir.

3.2.1. Mathematica

Mathematica yazılımı 1.0 versiyonu ile, Stephen Wolfram (<http://www.stephenwolfram.com>, 2006) tarafından 1987 yılında kurulmuş olan

Wolfram Research şirketi tarafından, 23 Haziran 1988 tarihinde kullanıma açılmıştır. 20 yıllık bir bilimsel programlama (*scientific computing*) deneyiminin ürünü olan Mathematica, günümüzde sayıları 1 milyonu aşkın oldukça geniş bir kullanıcı yelpazesine sahiptir.

En son versiyonu olan 4.5 ile Mathematica, oldukça gelişmiş bir kullanıcı ara yüzü ile donatılmıştır. Bu biçimiyle yazılımın, üzerinde düğmeler (*buttons*), giriş kutuları (*input boxes*) gibi birçok görsel öğenin kullanılabileceği etkileşimli uygulamalar hazırlamaya uygun bir yapısı vardır. Yazılımın diğer bir güçlü yanı, oldukça iyi bir uygulama geliştirme sistemine sahip olmasıdır. Bu yönüyle Mathematica kodlarına, birçok bilimsel makalede rastlamak mümkündür.



3.2.2. Maple

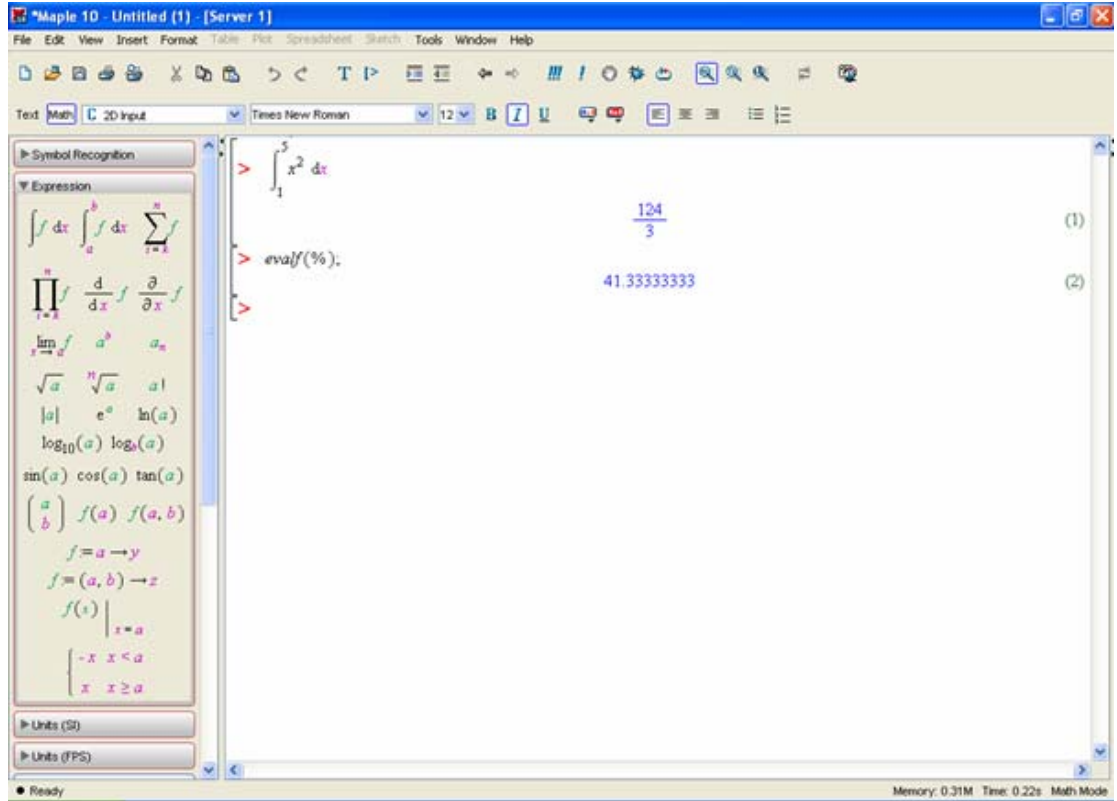
Maple, bilgisayar ile matematik çalışmalarında kullanılan en güçlü hesaplama sistemlerinden birisidir. Kullanım kolaylığı, genişleyebilirliği, işlem hızı ve minimum düzeyde bellek ve donanım kapasitesi gereksinimi ile Maple, son sürümü Maple10 olmak üzere, çeşitli düzenlemeleri ile 10 yılı aşkın bir süredir dünya üzerinde başta matematikçiler ve mühendisler olmak üzere 100.000'in üzerinde kayıtlı kullanıcı sayısına sahiptir.

Maple'in başlıca özellikleri arasında sayısal ve sembolik hesaplama, her türlü matematiksel notasyonu yazabilme, 2 ve 3 boyutlu grafik çizimleri ve grafik animasyonları sayılabilir. Bu özellikleri ile Maple, yoğunlukla analiz (calculus) ve diferensiyel denklemler olmak üzere geometri, lineer cebir, olasılık ve istatistik, ayrık matematik, sayılar teorisi ve nümerik analiz gibi matematiğin pek çok dalında etkin olarak kullanılabilir. Bunların yanı sıra, 2500 dolaylarında hazır matematiksel program, Maple'in program kütüphanesinde kullanılabilir durumdadır. Ayrıca Pascal benzeri yüksek-düzeyle bir programlama dili sayesinde amaca uygun olarak istenilen uygulamaların geliştirilmesi ve böylelikle kütüphanenin genişletilmesi mümkündür.

Maple, Waterloo Üniversitesinde 1980 yılının Aralık ayında Keith Geddes ve Gaston Gonnet tarafından kurulmuş olan *Symbolic Computation Group (SCG)* tarafından geliştirilmeye başlanmıştır. Bilgisayar Cebiri alanında birçok ispatlanmış teorem ve bunlar baz alınarak yazılmış bilimsel makalenin üzerine kurulan sistem, C programlama dili kullanılarak kodlanmıştır. Günümüzde Maple, Maple V Release 5, Maple6, Maple7, Maple8, Maple9 ve Maple10 sürümleri ile Macintosh, MS Windows, Unix, VMS, NeXT, Ultrix ve UNICOS gibi en popüler ve yaygın işletim sistemleri ortamlarında çalışabilmektedir. Maple çalışma sayfaları (worksheet) bu sistemlerin tümünde ortak bir görünüme sahip olduğundan, işlemler bir platformdan diğerine kolaylıkla taşınabilmektedir (Heck, 1996).

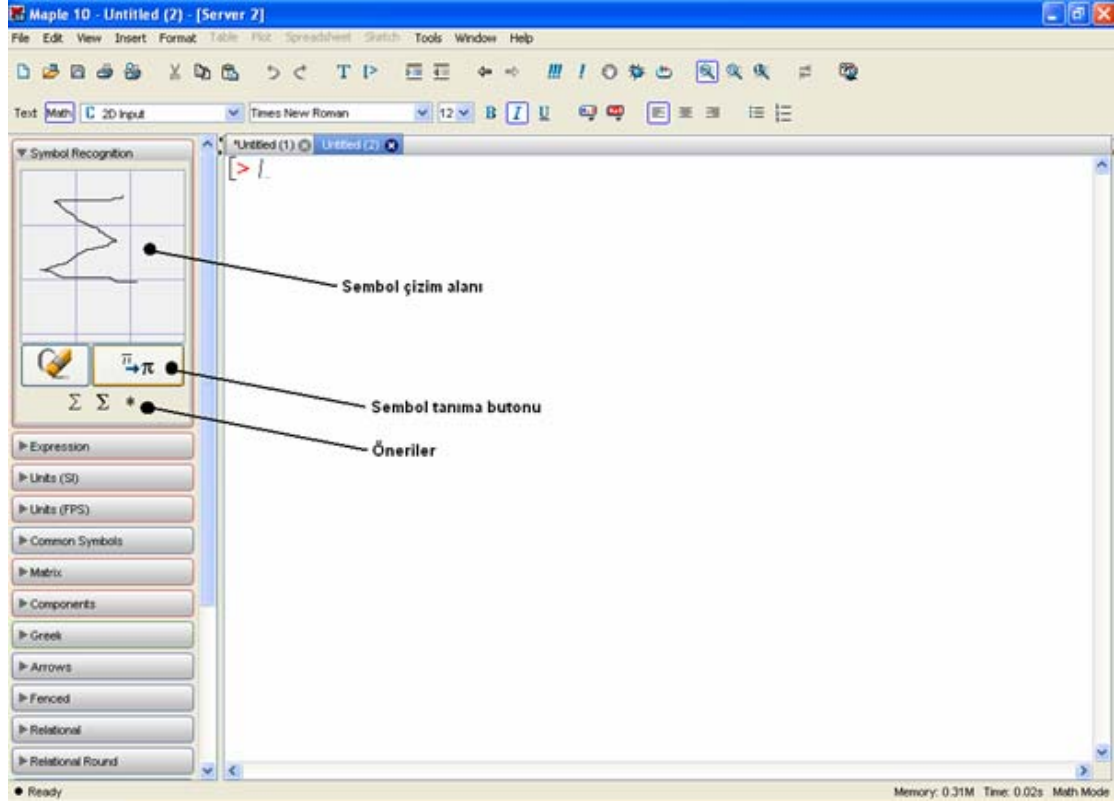
Maple serisi, yeni sürümü 10 ile, klasik "tek satır yazım formatlı" sisteme ek olarak, görsel nesnelerin kullanılabileceği yeni bir menü sistemi ve bu tür yazılımlar için bir ilk olarak nitelendirilebilecek olan "sembol tanıma sistemini" getirmiştir. Bu sistemde, sıklıkla kullanılan bazı hesaplamalar görsel menüden seçilerek gerçekleştirilmekte, bu sayede çalışma sayfasında komut-benzeri ifadeler yerine, geleneksel tarzdaki matematiksel ifadeler ve sonuçları yer almaktadır (Şekil 1).

Şekil 1
Maple 10 dan genel görünüm.



Sembol tanıma sisteminde ise, ilgili alana fare yardımıyla çizilen sembol, veritabanındaki tanınan matematiksel sembollerle bir benzetim algoritması kullanılarak karşılaştırılmakta, sonuçta eğer varsa eşleşen semboller önerilmektedir (Şekil 2).

Şekil 2
Maple 10 da şekil tanıma özelliği.



3.2.3. MathCad

Görsel yönden oldukça gelişmiş olan MathCad, teorisyenlerden çok mühendisler ve uygulamacılar arasında popüler olmuş bir yazılımdır. 1985 yılında kurulmuş olan MathSoft adlı firma tarafından geliştirilmiş olan yazılımın son sürümü olan MathCad 13, 2006 yılının ilk yarısında piyasaya sürülmüştür.



İki ve üç boyutlu modelleme konusunda başarılı olan yazılımın yeni sürümleri, XML, .NET, AVI gibi bazı standartlaşmış yazılım bileşenlerini destekleyerek, bu anlamda alana bir yenilik de getirmiştir.

1,5 milyonun üzerinde kayıtlı kullanıcısı olan yazılım (www.mathcad.com, 2006), adından da anlaşılacağı gibi (CAD-Computer Aided Design-Bilgisayar Destekli Tasarım) tasarıma yönelik birçok kütüphane bileşenine sahiptir ve bunların sayıları, kayıtlı kullanıcıları tarafından her geçen gün daha da artırılmaktadır.

3.3. Özel Amaçlı Bilgisayar Cebiri Sistemleri

Genel amaçlı bilgisayar cebiri sistemlerinin yanı sıra, matematik ya da fiziğin belirli bir alanında problem çözmek için geliştirilmiş bazı özel amaçlı yazılımlar da bulunmaktadır. Bunlara örnek olarak grup teorisi için geliştirilen Cayley (Butler vd., 1990) ve GAP (Schoenert vd., 1994), sayılar teorisi için geliştirilen PARI (Batut ve Canon, 1995), SIMATH (Hollinger ve Serf, 1991), ve KANT (Schmettow, 1991), komutatif cebir için geliştirilen CoCoA (Capani ve Niesi, 1995), cebirsel geometri için geliştirilen Macaulay (Stillman ve Bayer, 1989) ve Lie teorisi için geliştirilen LiE (Leeuwen vd., 1992) sayılabilir.

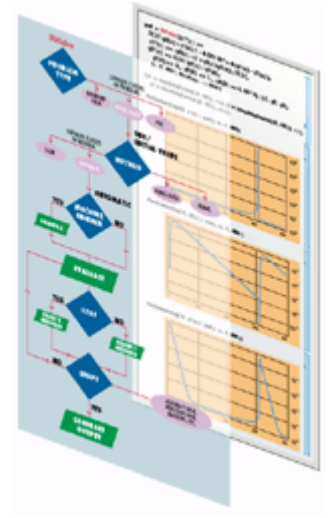
Özel amaçlı bilgisayar cebiri sistemleri arasında en çok öne çıkanlardan birisi olan GAP, GAP Group adlı çekirdek bir kadro ve tamamıyla gönüllülerden oluşan yüzlerce geliştiricisi ile, ticari amaçlar gütmeyen, internet üzerinde ücretsiz olarak dağıtılan bir cebir yazılımıdır (<http://www.gap-system.org/>, 2006). Adını “Groups, Algorithms, Programming” (Gruplar, Algoritmalar ve Programlama) kelimelerinden alan GAP, Hesaplanabilir Grup Teorisi (Computational Group Theory) disiplinin önemli bir ayağı olarak, Hesaplanabilir Sonlu Cebir (Computational Discrete Algebra)’in, programlama desteği de sağlayan bir sistemi olarak tanımlanmaktadır. Diğer özel amaçlı bilgisayar cebiri sistemlerinde olduğu gibi görselliği tümüyle

ikinci planda tutan yazılım, daha çok algoritmik yapısının gücüyle ve çok geniş bir geliştirme grubunun eseri olması sayesinde, minimum hata oranıyla dikkati çekmektedir.

3.4. Genel Amaçlı Bilgisayar Cebiri Sistemlerinin Bazı Ortak Özellikleri

3.4.1. Karar Verme

Kullanıcı tarafından özel bir çözüm metodu önerilmediği sürece yazılım en doğru ve en güvenilir metodu seçecek şekilde programlanmıştır. Ancak karar verme süreci, işlem zamanını artıracaktır. Bu sebeple, denklem çözümleri gibi bazı kritik işlemlerde, sisteme yardımcı olacak bazı yöntem önerilerinin kullanıcı tarafından yapılabilmesi olanaklı hale getirilmiştir.



Örneğin Maple 6'da,

$$ode14 := \left(\frac{\partial}{\partial x} y(x) \right) \left(\frac{\partial^4}{\partial x^4} y(x) \right) - \left(\frac{\partial^2}{\partial x^2} y(x) \right) \left(\frac{\partial^3}{\partial x^3} y(x) \right) + \left(\frac{\partial}{\partial x} y(x) \right)^3 \left(\frac{\partial^3}{\partial x^3} y(x) \right)$$

diferensiyel denklemini doğrudan **dsolve** komutunu kullanarak,

```
> dsolve(ode14);
```

biçiminde çözdürme işlemi güncel bir işlemciye sahip (P4, ~3 GHz) bir bilgisayarla ortalama 4 saniye sürmektedir.

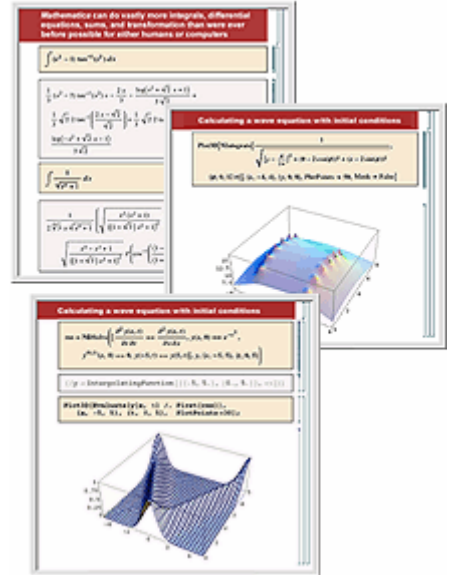
Oysa aynı hesaplamayı,

> dsolve(ode14, Lie);

biçiminde yaparsak, *Lie Simetriler Metodunu* (Cheb-Terrab vd. 1997) önermiş oluruz, ki bu işlemin ortalama süresi ise aynı konfigürasyonla 1,5 saniye sürmüştür.

3.4.2. Grafik Çizimleri

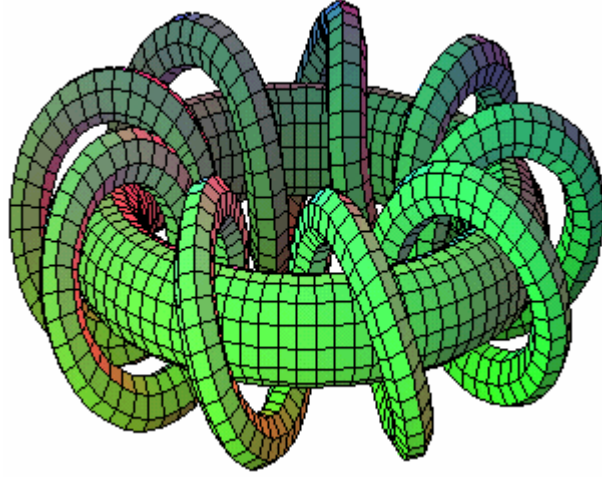
Bilgisayar cebiri sistemleri, özellikle Maple, Mathematica ve MathCad gibi görsel tasarıma sahip olanlar, çok gelişmiş 2 ve 3 boyutlu grafik çizme ve animasyon yeteneklerine sahiptirler. Grafikler, istenilen koordinat sisteminde, istenilen biçimde (noktasal ya da vektörel olarak) çizilebilmekte ve son derece hızlı bir şekilde çeşitli açılardan izlenebilmektedirler. Ayrıca *tubeplot* gibi bazı özel fonksiyonlar sayesinde, üç boyutlu fonksiyon grafiklerine ilgi çekici bir görsellik kazandırmak mümkündür (Şekil 3).



Bilgisayar ekranında matematiksel fonksiyonların grafiklerinin çizilmesi, iki ve üç boyutlu durumlar için ayrı ayrı değerlendirilmelidir. İki boyutlu durumda grafik, kabaca, belirli bir aralıkta, çoğunlukla yazılım tarafından belirlenen bir ölçek için y değerlerinin hesaplanması ve (x,y) koordinatlarının ekran düzleminde piksel renklendirme esasına göre işaretlenmesi ilkesine dayanır.

Üç boyutlu durumda ise, x ve y skalasına göre hesaplanan z değerleri ile oluşturulan (x,y,z) üçlülere, bir projeksiyon dönüşümü kullanılarak ekranda izdüşüm noktaları olarak belirlenirler.

Şekil 3
Maple’da bir *tubeplot* örneği.



3.4.3. Matematiksel Yazım Formatı

Tek-satır yazım formatında girilen matematiksel ifadeler, bilgisayar cebiri sistemlerinde oldukça başarılı bir biçimde klasik formata dönüştürülebilir.

$$\sum_{\mu=0}^{\infty} \frac{\varphi^{\mu} \cos\left(\frac{\pi\mu}{4}\right)}{\mu!^2 (\mu^2 + \kappa) (\mu^2 - \lambda)}$$

Genel olarak hemen hemen bütün disiplinlerde kullanılan semboller, bu yazılımlar tarafından desteklenmektedirler. Örneğin Mathematica’da 700’den fazla özel sembol kullanılabilir durumdadır. Bazı yazılımlar yeni semboller tasarlanmasına da olanak vermektedirler.

Ayrıca Maple 10 gibi bazı yeni sürümler, klasik tarzda (textbook style) veri girişini de olanaklı hale getirmişlerdir.

3.4.4. Uygulamalar

Bilgisayar cebiri sistemleri, script-tarzı programlama dilleri ve altprogram yapılarını kullanarak her türlü matematiksel uygulamayı geliştirmeye olanak verirler. Geliştirilen bu uygulamalar, bazı yazılımların herkese açık web sitelerinde karşılıksız olarak yayınlanırlar (<http://www.maplesoft.com> ve <http://www.mapleapps.com>, 2006). Uygulamaları genel olarak,

- Eğitim amaçlı uygulamalar,
- Belirli bir problemi çözmeye yönelik uygulamalar,
- Bilimsel bir araştırmada kullanılacak özel uygulamalar

şeklinde sınıflandırabiliriz.

3.4.5. Program Kütüphaneleri

Çeşitli bilimsel makaleler temel alınarak geliştirilen, ya da dünyanın her yerinden amatör ya da profesyonel çeşitli kullanıcıların geliştirdikleri yeni uygulamalar, bilgisayar cebiri sistemlerinin program kütüphanelerine dâhil edilirler. Bu kütüphaneler, zaman zaman yeni sürümlerle birlikte kullanıma açılırlar (MapleV Share Library gibi).

3.4.6. Karşılaşılan Sorunlar

Bilgisayar cebiri sistemleri, genel olarak üç sebepten dolayı kusursuz olarak nitelendirilemezler.

Bunlardan ilki, yan koşulların belirlenmesindeki eksikliktir. Gerek Maple, gerekse Mathematica,

$$\int x^a dx = ?$$

sorusuna aynı yanıtı vermektedir: $\frac{x^{(a+1)}}{a+1}$ Ancak her ikisi de $a \neq -1$ yan koşulunu bu cevaba eklememektedirler.

Benzer şekilde Mathcad,

$$\sum_{n=0}^{\infty} x^n \text{ ifadesini } \frac{-1}{x-1} \text{ şeklinde sadeleştirmekte, ancak bu eşitliğin sağlanması}$$

için gereken $|x| < 1$ koşulunu göz ardı etmektedir.

Diğer bir sorun, *bug* adı verilen programatik hatalardır. Bunlar sürümler yenilendikçe kullanıcıların bildirimleri doğrultusunda düzeltilseler de her sürüm, yeni hataları içermeye devam edecektir. Buna Maple'dan bir örnek verebiliriz:

$$1 - \frac{1}{2 + e^{3\pi ix}} = \frac{1 + e^{3\pi ix}}{2 + e^{3\pi ix}}$$

olduğu açıktır. Ancak MapleV Release 5 versiyonu $\int_0^2 \left(1 - \frac{1}{2 + e^{3\pi ix}}\right) dx$ ve

$\int_0^2 \frac{1 + e^{3\pi ix}}{2 + e^{3\pi ix}} dx$ belirli integrallerinin değerlerini sırasıyla 2 ve 0 olarak hesaplamaktadır.

Buna karşın Maple6, her iki integralin değerini de 1 olarak hesaplamaktadır.

Son sürümlerde belirli bir ölçüde giderilmesine karşın, üçüncü sorun olarak varsayımlar söylenebilir.

Özellikle sadeleştirme işlemlerinde sorun çıkaran varsayım sorununa örnek olarak, $\frac{\sqrt{u^2}}{u}$ ifadesini alabiliriz. Mathematica'dan bu ifadeyi sadeleştirilmesi istendiğinde, ifadeyi aynen bırakmaktadır. Oysa $u>0$ ise bu ifadenin sadeleşmiş biçimi 1'e eşit olmalıdır.

Öte yandan aynı işleme Maple6'nın yanıtı,

```
> simplify(sqrt(u^2)/u);
      csgn(u)
```

biçiminde olacaktır. Burada “csgn” fonksiyonu, “sgn” işaret fonksiyonunun karmaşık sayılara genişletilmişidir.

Ayrıca yine Maple6'ya eğer *assume* varsayım bildirimi kullanarak aynı soruyu sorarsak,

```
> assume(u>0);
> simplify(sqrt(u^2)/u);
      1
```

biçiminde doğru yanıtı alabiliriz.

Bu konuda Mitic ve Thomas (1994) şu ilginç saptamayı yapmışlardır: “Eğer ne yaptığınızı ve nasıl yapıldığını iyi biliyorsanız, bilgisayar cebiri sistemlerini kullanın; aksi halde kullanmayın.”

3.5. Maple’da Temel Kavramlar

Matematiğin yapıtaşlarını oluşturan sayılar, Maple’da temel veri türleri olarak karşımıza çıkmaktadır. Tamsayılar, matematikte algılandığı şekliyle tanımlanırlar ve programlama dillerinde olduğu gibi çok küçük sınırlar içersinde kalmazlar. Onbinlerce basamaklı sayılarla işlem yapılabilir. Örneğin Maple9, 10.000 tamsayısının faktöriyelini 35.662 basamaklı bir tamsayı olarak elde edebilmektedir. Özel sembolik metotlar sayesinde programlama dillerine göre sağladıkları bu üstünlük, bilgisayar cebiri sistemlerinin en önemli hesaplama özelliklerindendir.

Benzer şekilde, çok büyük ondalık sayılarla da işlem yapılabilir. Ancak Maple, ara işlemlerde ve sonuçlarda, aksi istenmediği sürece rasyonel ve irrasyonel sayıların ondalık açılımlarını vermeyecek, bu sayıları oldukları şekliyle bırakacaktır. Buna örnek olarak aşağıdaki çalışmayı verebiliriz:

```
> 2*sqrt(2);
```

$$2\sqrt{2}$$

Bu örnek, matematiksel ifadelere sayısal hesaplama ve sembolik hesaplama yaklaşımları arasındaki farkı açık olarak sergilemektedir. $\sqrt{2}$, sonsuz ondalık basamağa sahip irrasyonel bir sayıdır. Bu sayı bilgisayar tarafından sayısal olarak ancak sonlu bir ondalık basamağa kadar olan kısmı ile hesaplanabilir. Oysa sembolik hesaplama, sayıyı $\sqrt{2}$ olarak bırakıp, işlemin devam eden adımlarında da bu sembolü koruyarak sonucun tam (sembolik) olmasını garantilemektedir. İstenildiği takdirde bu işlemin sayısal sonucu da hesaplanabilir:

```
> evalf( 2*sqrt(2));
```

$$2.828427124$$

Hesaplanan sayı aksi belirtilmediği sürece 10 basamaklıdır. Ancak hesaplanacak sayının basamak sayısı da belirtilebilir:

> evalf(2* $\sqrt{2}$,200);

2.8284271247461900976033774484193961571393437507538961463533594
 75981464956924214077700775068655283145470027692461824594049849
 67211170147442528824299419987166282644533185501118551159990100
 23055641211430

Maple’da aritmetik işlemler, tanımlı aritmetik operatörler kullanılarak tek-satır matematiksel yazım kuralına uygun girildiklerinde, eğer tam olarak hesaplanabiliyorsa, işlemin sonucu yazılır:

> 2*5+3^2+(5-3)/2;

20

Ancak aritmetik işlemin sonucu kesin olarak hesaplanamıyorsa, hesaplanamayan ifade aynen bırakılır:

> 3*Pi/2;

$\frac{3\pi}{2}$

Maple’da cebirsel denklemlerin çözümleri, sembolik ve nümerik olarak iki grupta ele alınmıştır. “solve” fonksiyonu, verilen denklemi sembolik olarak çözmeye çalışacaktır.

> solve(x^3-1=0,x);

$1, -\frac{1}{2} + \frac{1}{2}I\sqrt{3}, -\frac{1}{2} - \frac{1}{2}I\sqrt{3}$

Ayrıca, sembolik hesaplamanın önemli avantajlarından birisi de, denklem çözümlerinde, aranan çözümün denklemdeki bilinmeyenlere bağlı olarak parametrik formda elde edilebilmesidir:

> $x := \text{solve}(y=m*x+c, x);$

$$x := \frac{-y+c}{m}$$

Daha karmaşık bir örnek olarak denklem sistemlerinin çözümü verilebilir. $y=mx+c$ doğrusu ile $x^2+y^2=r^2$ çemberinin kesim noktaları m , c ve r parametrelerine bağlı olarak hesaplatılabilir:

> $x := 'x':$

> $\text{Dogru} := y=m*x+c:$

> $\text{Cember} := x^2 + y^2 = r^2:$

> $\text{allvalues}(\text{solve}(\{\text{Dogru}, \text{Cember}\}, \{x, y\}));$

$$\left\{ x = \frac{1 - 2mc + 2\sqrt{-c^2 + r^2 + m^2 r^2}}{1 + m^2}, y = \frac{1}{2} \frac{m(-2mc + 2\sqrt{-c^2 + r^2 + m^2 r^2})}{1 + m^2} + c \right\}$$

$$\left\{ x = \frac{1 - 2mc - 2\sqrt{-c^2 + r^2 + m^2 r^2}}{1 + m^2}, y = \frac{1}{2} \frac{m(-2mc - 2\sqrt{-c^2 + r^2 + m^2 r^2})}{1 + m^2} + c \right\}$$

Öte yandan, sembolik olarak hesaplanamayan denklemlerin nümerik yöntemlerle hesaplanmaları mümkündür. Bunun için kullanılacak fonksiyon ise fsolve komutudur:

> $\text{fsolve}(\tan(\sin(x))=1, x);$

0.9033391108

> $\text{poly} := 23*x^5 + 105*x^4 - 10*x^2 + 17*x:$

> $\text{fsolve}(\text{poly}, x, -1..1);$

-.6371813185, 0.

3.6. Sembolik Hesaplama Kuramının Temelleri

Matematiksel ifadelerin (formüller, denklemler, fonksiyonlar vb.) gösterimi ve bu ifadeler üzerinde işlem yapmak, sembolik hesaplama kuramının en önemli temel taşlarından birisidir. Tek satır matematiksel yazım kuralına göre yazılmış bir matematiksel ifade üzerinde sembolik olarak doğrudan işlem yapmak olanaksızdır (Davenport vd., 1993). Dolayısıyla, öncelikle matematiksel ifadelerin ikili ağaç gösterimi ya da düğümsel matematik yapı olarak adlandırılan bir forma dönüştürülmeleri gerekmektedir. Bunu bir örnek üzerinde inceleyelim:

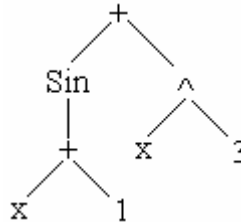
Örnek 1. $\sin(x+1)+x^3$ ifadesi, bu biçimiyle sembolik hesaplama açısından bir anlam taşımaz. Ancak aynı ifade,

$$+(\sin(+ (x,1)), ^{(x,3)})$$

biçiminde düğümsel matematik yapı olarak üzerinde her türlü sembolik hesaplamanın yapılabileceği bir matematiksel nesnedir. Aynı yapının ikili ağaç gösterimi Şekil 4’de verilmiştir.

Şekil 4

Matematiksel ifadelerin ikili ağaç gösterimi.

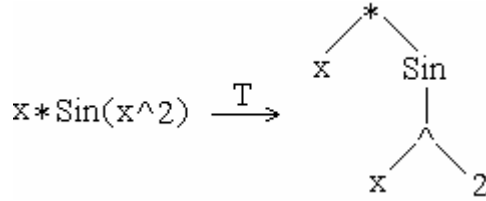


Düğümsel matematik yapılarıya dönüştürülen matematiksel ifadeler üzerinde, genel olarak bu ifadeler üzerinde konulacak hesaplama kuralları kullanılarak

sembolik hesaplama işlemleri gerçekleştirilebilir. Bu kurallar, işlemin niteliğine göre belirlenir.

Sözelimi, gerçekleştirilmek istenen hesaplama işlemi türev alma ise, bilinen türev alma kuralları genel düğümsel matematik yapılar üzerinde konulur ve bu kurallar ele alınan probleme uygulanır. Bu işlemleri, bir türev hesabı problemi örneği üzerinde gösterelim:

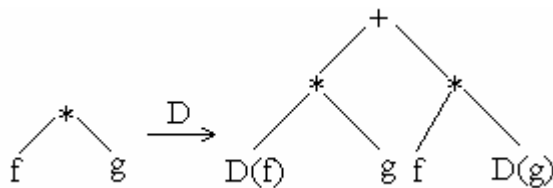
Örnek 2. $\frac{d}{dx}(x \sin(x^2)) = ?$ problemini göz önüne alalım. Öncelikle türevi hesaplanacak fonksiyonun düğümsel matematiksel yapıya dönüştürülmesi gerekmektedir. Bu dönüşümü gerçekleştirecek dönüşüme T dönüşümü diyelim. Bu durumda, ele alınan fonksiyona T dönüşümü uygulanırsa,



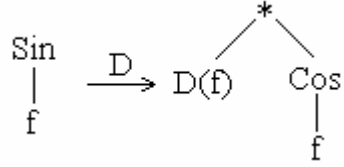
elde edilir. Burada, kuralların uygulanışı daha açık olarak izlenebildiğinden, soyut bir kavram olan ikili ağaç gösterimi kullanılmıştır.

Problemin çözülebilmesi için, sırasıyla *, Sin ve ^ düğümleri için türev kurallarına gereksinim vardır. Bu kuralları bir D dönüşümü yardımıyla tanımlarsak, ikili ağaç gösterimleri aşağıdaki gibi olacaktır:

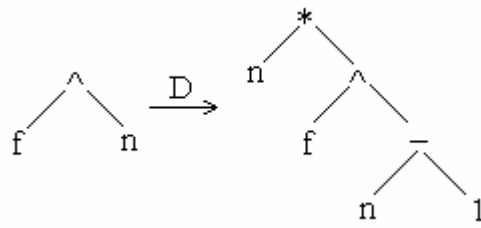
1. * Düğümü için türev kuralı:



2. Sin Düğümü için türev kuralı:

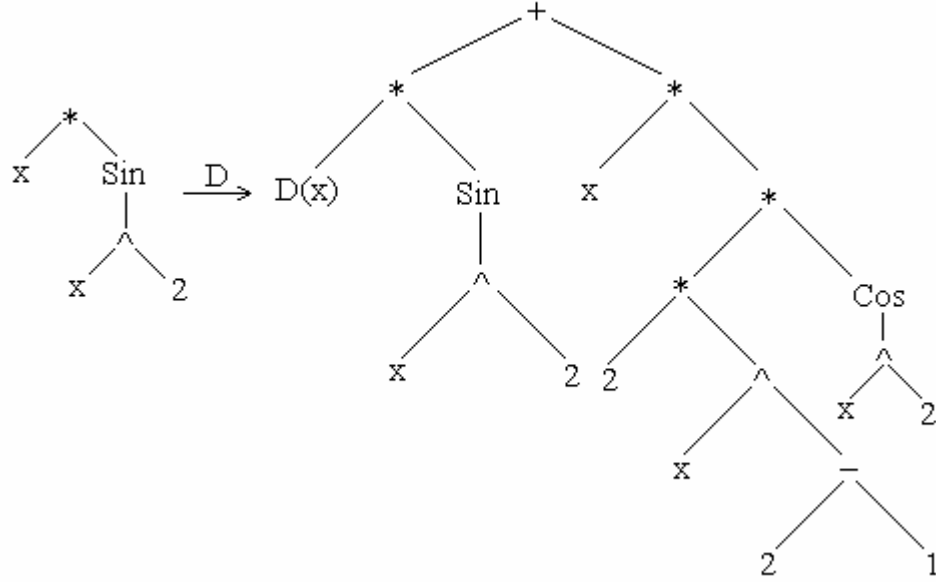


3. ^ Düğümü için türev kuralı:



Bu kurallar daha önce düğümsel matematik yapıya dönüştürülen fonksiyona uygulanırsa, Şekil 5 deki gibi türev fonksiyonunun ikili ağaç gösterimi elde edilmiş olur.

Şekil 5
 $x \sin(x^2)$ fonksiyonunun türevi için ikili ağaç gösterimi.



Bu yapıya T dönüşümünün tersi, yani T^{-1} dönüşümünü uygulayarak,

$$D(x) * \sin(x^2) + x * 2 * x^{(2-1)} * \cos(x^2)$$

şeklinde, türev fonksiyonunun ilk formu elde edilir. Bu fonksiyon üzerine, tanımlanacak sadeleştirme ve kısaltma kuralları da uygulanırsa, son şekliyle türev fonksiyonu,

$$\sin(x^2) + 2 * x^2 * \cos(x^2)$$

olarak elde edilir.

3.7. Sembolik Algoritmalar

Sembolik algoritmaların geliştirilme süreçlerini aşağıdaki gibi özetleyebiliriz:

1. Problemin tanımlanması.

Çözümü için algoritma geliştirilecek problem, girdi (input) ve çıktı (output) verileri, başlangıç atamaları (initialize) ile tanımlanır.

2. Algoritmanın yazılması.

Algoritmanın yazılması, standartlaşmış kurallar dâhilinde değildir. Genel olarak, koşullar, döngüsel yapılar ve atamalar açık bir şekilde belirtilerek anlaşılabilir bir dil kullanılmalıdır.

3. Algoritmanın sonlu bir adımda duracağını kanıtlanması.

Algoritmanın yapısı, aşağıdaki genel biçimlerden birisi olacaktır:

a. Algoritma iterasyon (tekrarlı işlem) yapısında değildir.

Bu durumda algoritmanın sonlu bir adımda duracağı açık olduğundan, kanıt gerektirmez.

b. Algoritma iterasyon (tekrarlı işlem) yapısındadır.

i. İterasyon sonlu döngü biçiminde

Bu durumda da algoritmanın döngüsel olarak sonlu bir adımda duracağı aşîkârdır.

ii. İterasyon koşullu döngü biçimindedir.

Bu durumda algoritmanın sonlu bir adımda duracağı kanıtlanmalıdır. Bu kanıtlama, aynı zamanda problemin çözümünde de önemli bir rol oynayacaktır.

4. Algoritmanın, durduğunda ürettiği sonuç ya da sonuçların doğruluğunun kanıtlanması.

Yine, problemin çözümüyle doğrudan bağı bir adımdır. Bu kanıt, aynı zamanda problemin çözümünün doğruluğunu gösterecektir.

5. Algoritmanın aynı girdi ya da girdilerle her çalıştırılışında aynı sonuç ya da sonuçları ürettiğinin kanıtlanması.

Problemin çözümünün tekliğini garantileyen adımdır.

6. Algoritmanın bir bilgisayar cebiri sisteminde (sembolik hesaplama programında) kodlanması.

Örnek 3. Collatz Problemi (Dolu Tanesi Sayıları). Bu problem, girdi verisi olarak pozitif bir tamsayıyı alarak, bu sayıyı başlangıç elemanı yapan bir dizinin tanımlanması esasına dayanır. Tanımlanan dizinin özelliği, her elemanın iteratif olarak kendisinden önceki eleman kullanılarak elde edilmesi, ve dizinin seçilen başlangıç sayısından bağımsız olarak sonunda her zaman 4,2,1,4,2,1,... kısır döngüsüne indirgenmesi savıdır. Henüz kanıtlanamamış olan bu savın çözümü, kurulacak algoritmanın sonlu bir adımda duracağını kanıtlanması ile denk olacaktır.

Problemin ifadesi aşağıdaki gibidir:

$$a_0 \in \mathbb{Z}^+$$

$$a_n = \begin{cases} 3a_{n-1} + 1, & a_{n-1} \text{ tek ise} \\ \frac{a_{n-1}}{2}, & a_{n-1} \text{ çift ise} \end{cases}$$

Örneğin,

$a_0=7$ için,

22,11,34,17,52,26,13,40,20,10,5,16,8,4,2,1, ...

$a_0=9$ için,

28,14,7,22,11,34,17,52,26,13,40,20,10,5,16,8,4,2,1, ...

olur.

Algoritma 1.

Girdi : $a_0 \in \mathbb{Z}^+$

Çıktı : $(a[1], a[2], \dots)$

Başla

$k \leftarrow 0$

$a[k] \leftarrow a_0$

Tekrarla

$k \leftarrow k + 1$

Eğer $a[k-1]$ tek ise;

$a[k] \leftarrow 3 \times a[k-1] + 1$

Aksi halde

$a[k] \leftarrow a[k-1] / 2$

$a[k] = 1$ olana kadar

Bitir.

BÖLÜM 4

MATEMATİK ÖĞRETİMİ VE BİLGİSAYAR

Tamamıyla matematiksel bir altyapıya sahip olsa da bilgisayarlar, matematiksel anlamda çok yakın bir zamana kadar sadece bazı mühendislik problemlerinde karşılaşılan sayısal hesaplamalar için kullanılmaktaydı. Bunun sebebi, bilgisayarların donanım yapılarındaki az gelişmişlik ya da bilgisayarların temel mantığını oluşturan teorilerde ortaya çıkan yeni yaklaşımlar değildir. Çünkü bilgisayarlar, 1854 yılında George Boole tarafından geliştirilen ve 0 ve 1 den oluşan ikili sayı sistemine göre inşa edilmiş, Boolean Cebiri adı verilen cebirsel yapı esas alınarak tasarlanmış makinelerdir. Temelde aynı prensibe göre çalışan hâlihazırdaki bilgisayarların en büyük avantajı, yazılımlarda gerçekleşen önemli gelişmelerdir. Dolayısıyla bu gelişmeler, matematik programlarında da son yıllarda bazı yeni yaklaşımların oluşmasına sebep olmuştur. Bu gelişmelerden söz etmeden önce, matematiğin, gelişmelerinde oynadığı önemli rolü vurgulamak açısından, bilgisayarların tarihçesine kısaca bir göz atmanın faydası olacaktır.

4.1. Bilgisayarın Kısa Bir Tarihçesi

4.1.1. Donanım Gelişimi

Bilgisayar kavramının tarihçesi, insanoğlunun hesap yapmak amacıyla icat ettiği makinelerle başlamaktadır. Bunun sebebi, gün geçtikçe artan “hesap yapma” gereksinimini, elle ya da zihinsel yöntemlerle karşılayamamaları olabilir. Hesap yapma işlemini hızlandırmak için, otomatik bir mekanizmanın kullanılabileceği fikri M.Ö. 1000 yıllarında kadar dayanmaktadır. Bu yıllarda ilk kez Çinlilerin kullandığı

kabul edilen Abaküs, bu anlamda ilk mekanik hesaplayıcı, dolayısıyla da bilgisayarın atası olarak görülebilir.

Hesap yapmak amacıyla geliştirilen bu makinelerin bilgisayara dönüşümü hikâyesini kronolojik olarak aşağıdaki gibi verebiliriz:

Pascalline: Hesap makinesi sayılabilecek ilk ciddi icat Fransız matematikçi Blaise Pascal tarafından geliştirilmiştir. Babası vergi dairesinde memur olarak çalışan Pascal, On altı yaşında iken, 1642 yılında “Pascalline” adlı mekanik hesap makinesini icat etmiştir.



Değişik sayıda dişleri olan çarklardan oluşan bu makine, sadece toplama ve çıkarma işlemlerini otomatik olarak yapabilmektedir.

Leibniz Çarkı: Alman matematikçi Gottfried Wilhelm Leibniz, Pascal’ın 1642 yılında geliştirdiği hesaplayıcının fonksiyonlarını daha da artırarak, 1671 yılında “Leibniz Çarkı” adlı aygıtı icat etmiştir (Şekil 6). Bu aygıt, toplama ve çıkarma işlemlerinin yanı sıra bölme, çarpma ve karekök alma işlemlerini de yapabiliyordu.

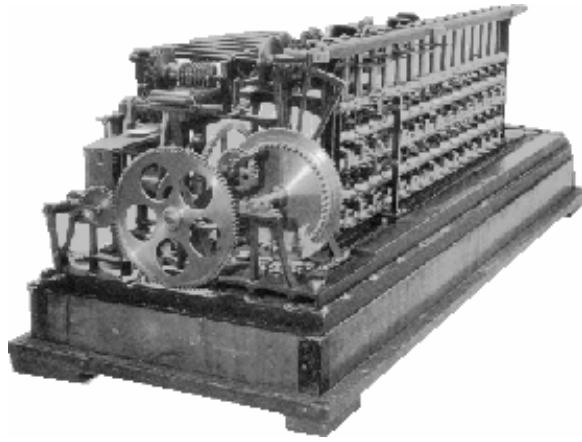
Şekil 6
Leibniz çarkı



Fark Makinesi: Bu konuda ciddi çalışmaları olan bir diğer kişide Charles Babbage’dır (1791-1871). Babbage, matematiksel işlemlerin yanı sıra, günlük hayatta karşılaşılan bazı problemleri çözmeye yönelik olarak başka işlemleri de

yapabilen bir makine tasarlamayı planlıyordu. İngiliz hükümetinin parasal desteğini de alan Babbage, uzun süren çalışmalar sonunda “Fark Makinesi”ni 1830 yılında icat etti (Şekil 7).

Şekil 7
Babbage’ın fark makinesi



Babbage, daha sonra “Analitik Makine” adını verdiği, buhar gücü kullanarak otomatik olarak çalıştırılacak ve diğer hesaplayıcılardan daha fazla fonksiyona sahip olacak bir proje üzerinde çalışmaya başladı. Babbage, bu projeden istediği neticeyi alamadan 1871 yılında hayatını kaybetti ve dolayısıyla proje yarım kaldı. Babbage analitik makinesinde, bugünkü bilgisayarların temel prensibine benzer olarak, bir mantıksal işlem birimi, bir veri depolama birimi ve giriş/çıkış (input/output) birimleri kullanmayı planlıyordu. Bu sebepten, Babbage, modern bilgisayarın babası olarak da anılmaktadır.

Augusta Ada Lovelace (1816-1852), Analitik Makine prensibinde Babbage ile beraber çalışmış ve ona yardımcı olmuştur. 1842 yılında yazdığı notlarında Analitik Makine kullanılarak hazırlanacak programlar ile insanlar tarafından çözüm

yolu bilinen, ancak işlem fazlalığı sebebiyle çözülemeyen birçok problemin kolaylıkla çözülebileceğini anlatmıştır. Lovelace, 36 yaşındaki erken ölümü ile arkasında bıraktığı bu notlarla çok değerli fikirlerini takipçilerine aktarmıştır.

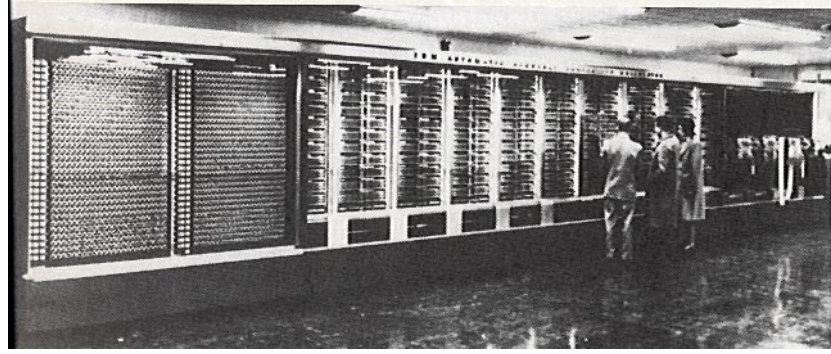
Mark-I: Alman asıllı bir Amerikalı olan istatistikçi Herman Hollerith (1860-1929), 1890 yılı nüfus sayımında delikli kart sistemini kullanan, kendi geliştirdiği bu makineyi kullanmıştır. Mark-I ile sayımın değerlendirme süresi dörtte bire düşmüştür.

Hollerith'in geliştirdiği makine aslında, Joseph Maria Jaquard'ın (1752-1834) 1806 yılında tasarladığı kart sistemini kullanıyordu. Mark-I, kartları verilen kodlara göre delerek bilgiyi depoluyor, delikli karttaki bilgiyi tekrar okuyabiliyor ve bu bilgiyi kullanabiliyordu. Geliştirilen makinenin daha öncekilere göre üstünlükleri olmasına karşın, istenilen hız ve doğru sonuca ulaşamamıştı. Mark-I, insan müdahalesine gereksinim duyduğundan, yarı-otomatik olarak nitelendirilmektedir.

Hollerith, makinesinde yaptığı geliştirmelerle seri üretime 1896 yılında kurduğu “Tabulating Machine Company” adlı bir şirket ile devam etmiştir. İleride bu firma, daha çok ofis araçları ve saat üreten başka bir firma ile birleşerek IBM (International Business Machines) ismini alacaktır ve günümüze kadar bilgisayar dünyasının en önemli şirketi olma unvanını koruyacaktır.

Mark-Ic: Harvard Üniversitesinden Howard Hattaway Aiken'in (1900-1973) tasarladığı ASCC (Automatic Sequence Controlled Calculator-Otomatik Dizi Denetimli Hesap Makinesi) projesi, IBM firması ile gerçekleştirdiği ortak çalışmalar sonucunda 1944 yılında Mark-Ic adlı makinenin üretilmesiyle sonuçlanmıştır. Mark-Ic saniyede 5 işlem yapabiliyordu ve 18 m uzunluğunda ve 2,5 m yüksekliğindedir. Logaritmik ve trigonometrik fonksiyonlarla bile işlem yapabilen bu makine, insan müdahalesi olmadan hazırlanan programı yürüten ilk bilgisayar olmuştur. Bununla birlikte Mark-Ic, elektronik bir bilgisayar değildir; elektromekanik olarak sınıflanmaktadır (Şekil 8).

Şekil 8
Mark-Ic



ENIAC: Mark-Ic den kısa bir süre sonra, Pensilvanya Üniversitesinden John William Mauchly (1907-1980) ile John Presper Eckert (1919-1995), ENIAC (Electronic Numerical Integrator And Calculator-Elektronik Sayısal Bütünleştirici ve Hesaplayıcı) isimli sayısal elektronik bilgisayarı 1946 yılında tamamlamışlardır (Şekil 9).

Yapımında 18,000 adet elektronik tüp kullanılan ENIAC, 150 kwatt gücünde idi ve 50 ton ağırlığıyla 167 m² yer kaplıyordu. Saniyede 5000 toplama işlemi yapabiliyordu ki bu da Mark-Ic'den 1000 kat daha hızlı olduğu anlamına geliyordu. ENIAC, yalnızca 80 karaktere eş veri saklayabiliyordu ve lambaların hepsi çalıştığında, mühendis ekibi bir problemi çözmek, için 6000 kabloyu elle fişe takarak ENIAC'ı kurmaktalardı.

Şekil 9
ENIAC'dan çeşitli görüntüler.



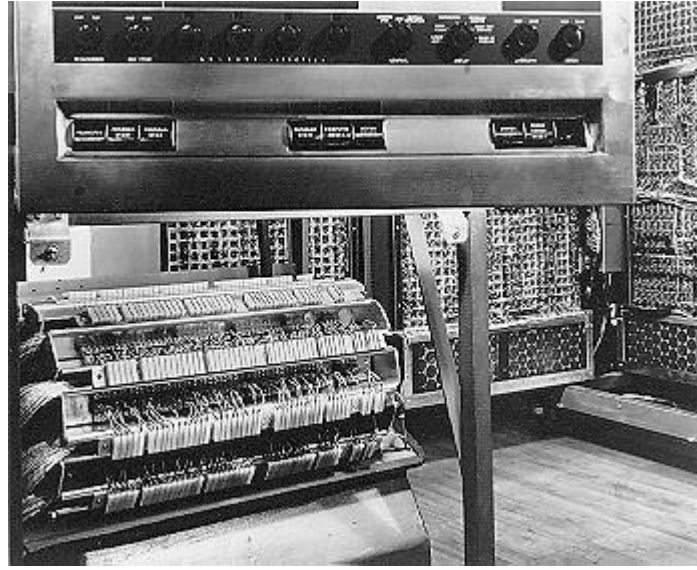
Ne yazık ki bir çok önemli bilimsel gelişmenin motivasyon kaynağında olduğu gibi, ENIAC projesinin temelinde de askeri amaçlar yer almaktadır. Bilgisayar, uzunca bir süre top mermilerinin menzillerinin hesaplanması gibi problemlerin çözümünde kullanılmıştır.

EDVAC: ENIAC projesinin, yine Eckert & Mauchly ikilisi tarafından, matematikçi John Von Neumann'ın (1903-1957) 1945 yılında yayınladığı bir raporda yer alan görüşleri doğrultusunda geliştirilmesi sonucunda, 1946 yılında Amerikan ordusuna ait “Balistik Araştırmalar Laboratuvarı”nda EDVAC (Electronic Discrete Variable Automatic Computer-Elektronik Ayrık Değişken Otomatik Bilgisayar) adlı yeni bir bilgisayar üretilmiştir (Şekil 10). Bu bilgisayar, geliştirilen yeni bellek teknolojisi sayesinde ENIAC'dan on defa daha küçüktür ve ENIAC'ın yüz katı kadar daha hızlı çalışabilmektedir. EDVAC, kendisinin getirdiği bir yenilik olarak komutların da diğer veriler gibi bilgisayara dışarıdan girilebilmesini sağlıyordu. Bu özellik,

programcılıkta büyük kolaylıklar sağlamış, yazılım teknolojisinin gelişimini hızlandırmıştır. Bir anlamda, bu günkü şekliyle “programlanabilir” ilk bilgisayar olma özelliğine sahiptir.

Şekil 10

EDVAC

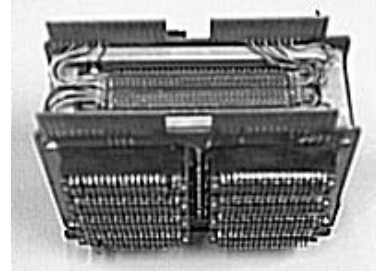


UNIVAC: EDVAC’dan sonraki gelişme, 1951 yılında yapılan UNIVAC isimli bilgisayardır. Aynı ekip tarafından geliştirilen UNIVAC, ilk defa manyetik teyp kullanarak verileri depolayan bilgisayar olma özelliğine sahiptir. Böylece günümüzde kullanılan manyetik disklerin (floppy disket ve sabit diskler gibi) temelleri atılmış, veri saklama işlemi “mekanik” sistemden “manyetik” sisteme terfi etmiştir. Ayrıca sistemin önceki bilgisayarlara göre önemli bir farkı, yeni ve gelişmiş bir bellek teknolojisine sahip olmasıdır (Şekil 11).

Şekil 11
(a) UNIVAC,
(b) UNIVAC sistemlerinde kullanılan gelişmiş bellek teknolojileri



(a)



(b)

IBM 700 Serisi: 1950’li yıllar, vakum tüplerinin birçok makinede yaygın olarak kullanılmaya başlandığı dönemlerdir. UNIVAC ve daha gelişmiş bir sistem olan IBM 700 serisi, vakum tüpler kullanılarak yapılan elektronik bilgisayarlardır. Ancak vakum tüplerinin en önemli dezavantajları, çok enerji harcamaları ve çok fazla ısınarak bilgisayarların sıklıkla arıza yapmasına sebep olmalarıdır. Tüplerin boyutlarının büyük olması ise, bilgisayarların daha da küçülmesi önünde başka bir sorun olarak ortaya çıkmaktadır.

Bu yıllarda program yazabilmek için, kullanılan bilgisayar donanımının çok iyi bilinmesi gerekmektedir. Zira program yazmak için makine dili kullanılmaktadır.

Philco Transac S-200 IBM 1401: Vakum tüplerine göre az enerji harcayan, daha az yer kaplayan ve fazla ısınmayan elektronik devre elemanları olan transistörlerin kullanılmaya başlandığı yıllarda bilgisayar dünyasında da önemli gelişmeler yaşanmıştır. Philco Transac S-200 IBM 1401, transistör kullanılarak üretilen ilk bilgisayardır.

IBM 360: 1960 yılından sonra, binlerce transistörü içerisinde bulundurabilen silikon entegre devreler üretilmeye başlanmış, entegrelerin kullanılması, bilgisayarların boyutlarının küçülmesini, maliyetlerinin azalmasını ve işlem hızlarının artmasını sağlamıştır. Bu yıllarda manyetik diskler üretilmiş, entegrelerin kullanımı ile merkezi işlem birimleri (CPU-Central Process Unit) üretilmeye başlanmıştır. IBM 360, entegre devre elemanının kullanıldığı ilk bilgisayarlardandır.

Intel 4004 Mikro İşlemcisi: 1970 yılından sonra entegre devre teknolojisi hızla gelişimine devam etmiş, entegre devreler birleştirilerek mikroçipler üretilmeye başlanmıştır. Intel 4004, entegrelerin birleştirilmesiyle oldukça hızlı hale gelen ilk merkezi işlem birimi sayılabilir.

Apple-I ve II: 1977 yılında piyasaya çıkan Apple, 1976 yılında iki üniversite öğrencisi, Steve Wozniak ve halen “Apple Computers Inc.” firmasının genel müdürü olan Steve Jobs tarafından, bir evin garajında üretilmiştir. İlk olarak Apple-I adıyla 1976 yılının Nisan ayı ile 1977 yılının Mart ayı arasında üretilen bilgisayar, tümleşik bir klavyeye sahipti ve monitörü bulunmuyordu.

Şekil 12
Apple-I bilgisayarı.



Merkezi işlemci olarak Intel yerine Motorola firmasının üretimi olan 6800 mikroçip modelini kullanmayı tercih eden Apple-1, 4 kilobyte belleğe sahiptir ve herhangi bir monitör sistemine ya da televizyona, grafik içermeyen metin-tabanlı görüntü gönderebilmekte, ancak ses ya da renk üretimi yapamamaktadır.

1977 yılının Nisan ayı ile 1979 yılının Mayıs ayı arasında geliştirilen Apple-II ise, 6502 işlemcisine sahiptir ve başlangıç olarak 16 kilobyte belleğe sahiptir. Bu büyüklük, 1981 yılında üretilen modelde 48 kilobyte'a kadar çıkacaktır. Apple-II'nin kullanıcıları cezbeden en önemli özelliklerinden birisi renkli grafikler üretebilmesi, diğeri ise ROM belleğine kaydedilmiş dâhili BASIC programlama dili yorumcusu sayesinde, her kullanıcı tarafından kolaylıkla program yazılabilir hale gelmiş olmasıdır. Ayrıca sisteme eklenmiş, son derece kaliteli görüntü verebilen bir monitör sistemi de günümüz bilgisayarlarının yavaş yavaş şekillendiğinin habercisi niteliğindedir.



IBM PC: Günümüzde de söz sahibi olan IBM firması ilk kişisel bilgisayarını 1981 yılında piyasaya sürdü. Kısa bir zaman diliminde standart haline gelen IBM PC'lerin 4 yıl sonunda bir milyonuncusu satılmıştı. Artık dünyanın her tarafında IBM uyumlu bilgisayarlar üretilmeye başlanmış, yazılımların çoğunluğu da IBM PC uyumlu olarak yazılmaya başlanmıştır. İlk etapta üretimin büyük bir kısmı Amerika Birleşik Devletleri'nde gerçekleştirilirken, iş gücünün daha ucuz olması gibi bazı ekonomik nedenlerden zamanla üretim uzak doğu ülkelerine kaymıştır (Şekil 13).

Şekil 13

1981 yılının Ağustos ayında piyasaya sürülen ilk IBM PC bilgisayarı.



IBM, 1983 baharında, içinde sabit disk bulunan ilk kişisel bilgisayar olan IBM PC/XT modelini piyasaya sürmüştür. Sabit disk, yerleşik bir depolama aygıtı olarak çalışmaktadır ve 10 megabayt'lık bir kapasiteye sahiptir.

1984 yılında firma, Intel'in 80286 mikro işlemcisini içeren, IBM PC/AT adlı yüksek performanslı ikinci kuşak bilgisayarını tanıtmıştır. Bu bilgisayar, IBM PC'den üç kat daha hızlıdır (Şekil 14).

Şekil 14
80286 işlemcili IBM PC/AT.



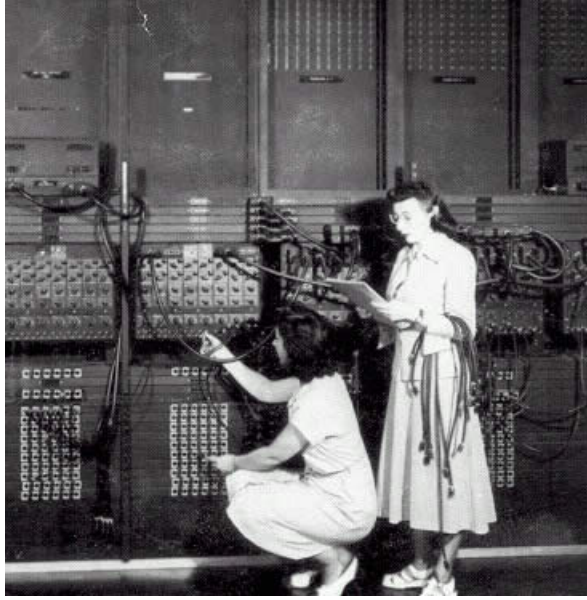
Bu dönemden günümüze kadar bilgisayar teknolojisi büyük bir hızla ilerlemiştir. Bilgisayarların donanımlarındaki gelişmelere paralel olarak gerçekleşen yazılım alanındaki gelişmeler, bilgisayarların sadece uzman kişilerin değil, bütün insanların kullanabileceği makinelere dönüşmesini sağlamış, dolayısıyla artık bilgisayar, insan hayatının ayrılmaz ve vazgeçilmez bir parçası olmuştur.

4.1.2. Yazılım Gelişimi

Donanım gelişimine paralel olarak süren yazılım gelişimi de, son on yılda kayda değer bir ivmelenme ile bilgisayarları, herhangi bir uzmanlık gerektirmeyen, her yaş ve eğitim seviyesinden kullanıcının kolaylıkla kullanabileceği makineler haline getirmiştir. Kuşkusuz ki bunda en önemli pay, bilgisayarın açıldığı andan itibaren bütün birimlerinin çalışmalarını kontrol eden temel yazılım olan işletim sistemlerindeki gelişmelere aittir. Bunun yanı sıra, program yazma, diğer bir deyişle genel amaçlı olarak üretilen bilgisayarları kendi isteklerimiz doğrultusunda çalışabilir biçime getirme, ya da elle çözülmesi işlem fazlalığı açısından çok zor veya imkânsız olan bir problemi bilgisayar desteğinde çözme işlemi, geçmişteki bilgisayarlarda oldukça zor ve zahmetli bir işlem iken, günümüzde, öğrenilmesi çok kolay programlama dilleri ile oldukça basite indirgenmiştir (Şekil 15).

Şekil 15

ENIAC'da program yazan iki mühendis.



Mekanik programlamanın ardından, geliştirilen veri girişi teknolojileri sayesinde gerçekleştirilebilen makine dilinde programlama işlemi de, sadece 1 ve 0 rakamlarından oluştuğundan, ikili sayı sistemine oldukça hâkim olmayı gerektirmektedir. Bu süreçte geliştirilen ilk sözdizimsel programlama dili olarak kabul edilen “Assembler” programlama dili bu zorunluluğu ortadan kaldırmıştır; ancak hâlâ program yazma işlemi çok özel bir uzmanlık gerektiren alandır.

Programlama dillerinin gelişimi, 1970’li yıllarda hız kazanmış, özellikle 1969–1973 yılları arasında Bell Laboratuvarlarında Dennis Richey ve David Kerningham tarafından geliştirilen C programlama dili, daha sonra birçok programlama diline esin kaynağı olmuştur. Pascal ve C gibi yapısal dillerin ardından, yeni bir kuşak olarak görsel programlama dilleri geliştirilmiş ve sunucu/istemci mimarisi adı verilen yazılım geliştirme teknikleri uygulanmaya başlanmıştır. Günümüzde ise, yerel ağ yapıları üzerinde çalışan yazılımlar yerine, internet teknolojilerindeki gelişmelere bağlı olarak daha çok internet protokollerini kullanan

yazılımların geliştirilmesine olanak veren, Java, php, asp gibi programlama dilleri önem kazanmaya başlamıştır.

Programlama dillerindeki gelişmelere paralel olarak, işletim sistemleri de bir yandan Motorola işlemcili Apple bilgisayarların kullandığı Macintosh sistemleri olarak, diğer yandan da Intel tabanlı IBM uyumlu PC bilgisayarlar için önce IBM DOS, daha sonra buradan ayrılan Bill Gates ve arkadaşları tarafından kurulan Microsoft firması tarafından geliştirilen MS DOS işletim sistemleri olarak gelişmelerini sürdürmüşlerdir.

1984 yılında Macintosh tarafından ilk işaretleri verilen, üst üste açılan pencereler şeklinde çalışan “görsel işletim sistemi” mantığı, yazılımın 1985 yılında çıkarttığı işletim sistemi ile oldukça ses getirmiştir. “MacOS” adlı bu sistem, günümüzde kullanılan bir çok benzer işletim sistemi için esin kaynağı olması açısından, işletim sistemleri tarihinde önemli bir yere sahiptir.

1990 yılının Mayıs ayında Microsoft firması tarafından, bunun alternatifi olarak “MS Windows 3.0” ve ondan kısa bir süre sonra “MS Windows 3.1” piyasaya sürülmüştür. Ancak bunlardan sonra, bilgisayar ağları için özel olarak geliştirildiği iddia edilen “MS Windows 3.11 for Workgroups”, ve gerçek anlamda ağ desteği sağlayan “Windows NT” işletim sistemleri geliştirilmiştir. Bunlara paralel olarak, güvenlik ve hız gibi sebeplerden, daha çok kurumsal olarak tercih edilen Unix ve son zamanlarda popülaritesi giderek artan Linux işletim sistemleri de gelişmelerini sürdürmüşlerdir.

4.2. Bilgisayar Destekli Matematik Öğretimi

Matematik öğretiminde bilgisayarın rolü, 90’lı yıllarda, matematik eğitiminde görselliğin önemini vurgulayan çalışmalardan sonra oldukça artmıştır. Bu konuda anahtar rol oynayan çalışmalardan birisi, Dörfler (1993) tarafından gerçekleştirilmiştir. Sembolik ve cebirsel yapıların bilgisayar yardımı ile kolaylıkla

geometrik yapılara dönüştürülebileceğini söyleyen Dörfler, görsel nesnelerin formal tanımların yerlerini alamayacağını, ancak “anlaşılabilirliklerini” büyük ölçüde artıracığını vurgulamıştır.

Bunu destekleyen önemli bir çalışma da Dreyfus (1994) tarafından yapılan ve bilgisayarı bir “bilişsel araç” olarak vurgulayan çalışmadır. Dreyfus, teorik bilgi birikiminin bilgisayar yardımı ile görselleştirilebileceğini, böylece de mevcut bilgi birikiminin “ezbere bilgi” konumundan “öğrenilmiş bilgi” konumuna terfi edebileceğini, fraktal geometriyi kullanarak çarpıcı bir şekilde göstermiştir. Öyle ki, iteratif fonksiyonların görselleştirilmesi, bilgisayar kullanılmadığı takdirde hemen hemen imkânsız olmaktadır.

Bilgisayar destekli matematiğin en önemli ayaklarından biri olan bilgisayar cebiri sistemlerinin kullanılabilirliği ve güvenilirliği, 90’lı yılların başlarından günümüze kadar en çok tartışılan konuların başında yer almıştır. Kullanılabilirlik kavramı, genel olarak yazılımların görsel tasarımları, yardım dokümanlarının içeriği ve öğrenilmelerinin kolaylığı gibi özellikleri ile ilişkili olarak ortaya atılmış bir kavramdır ve bir yazılımın ya da internet sitesinin “ne kadar iyi” olduğunun belirlenmesinde önemli bir ölçek olarak kullanılmaktadır. Bilgisayar cebiri sistemlerini bu açıdan incelediğimizde, karnelerinin çok da parlak olmadığını gözlemliyoruz.

Bu konuda diğerlerine göre daha “kullanıcı dostu” (user-friendly) bir arayüze sahip olan Mathcad dışında, bilgisayar cebiri sistemleri bu özelliği çok da önemsememişlerdir. Maple, en son sürümüne kadar, kullanıcılarını sistemi kullanabilmeleri için en az bir programlama diline hâkim olmaya zorlayan bir arayüz tasarımı ısrar etmiştir. Daha önceki kesimlerde de anlatıldığı gibi, son sürümü (Maple 10) ile bu anlayışını biraz değiştirmiş ve kullanılabilirliğini bir ölçüde artırmıştır.

Başlangıçta geleneksel kullanıcı kitlesi olarak sadece temel bilimciler ile mühendisleri hedefleyen bu yazılımların, kolay ve herkes tarafından kullanılabilir

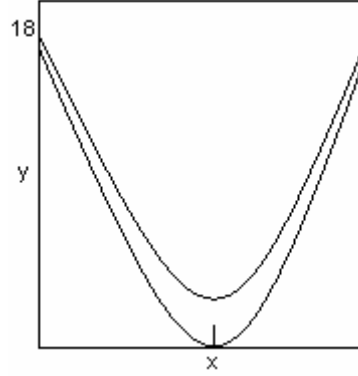
olmayı önemsememeleri doğal karşılanabilir. Çünkü bu sistemler, son yıllara kadar, hitap ettikleri kullanıcı yapısı itibarı ile zaten nümerik analiz konusunda, dolayısıyla da Fortran, Pascal ya da C gibi bir programlama dillerinde belirli bir bilgi birikimine ve beceriye sahip kullanıcılar tarafından kullanılmaktadırlar. Ancak son zamanlarda popülaritesi giderek artan bilgisayar destekli matematik öğretimi, bu sistemlerin eğitimde kullanılmalarını kaçınılmaz kılmıştır. Bu sebeple, hemen hemen bütün bilgisayar cebiri sistemleri, “eğitim” kategorisini internet sayfalarına eklemiş, bu konuda gerek kullanıcıları ile paylaşım yoluna giderek, gerekse kendi yazılım ekiplerini kullanarak, eğitimcilerle de işbirliği içerisinde yeni kütüphaneler ve bileşenler (package) geliştirmişlerdir. Bunların büyük bir kısmı da kullanıcılara ücretsiz olarak sunulmaktadır. Öyle ki, bu yazılımların internet sitelerinde daha önce görmeye alışık olmadığımız, orta öğretim, hatta ilköğretim seviyesinde çalışma sayfalarına ve etkinliklere rastlamak mümkün olmaktadır.

Kajler ve Soiffer (1994), Maple, Mathematica, Macsyma ve Mathcad yazılımlarını, “insan-bilgisayar etkileşimi” açısından inceledikleri çalışmalarında, bilgisayar cebiri sistemlerinin eğitimde kullanılabilirliklerine ilişkin önemli saptamalarda bulunmuşlardır. Kajler ve Soiffer’e göre bu sistemlerin eğitsel açıdan en önemli avantajları, elde edilen çıktıları “elle yazılışa” en yakın şekilde verebilmeleri, üniversite altındaki öğrencilerin pekte hoşlanmadıkları ondalık açılımlar yerine rasyonel veya bilinen irrasyonel sayıların (π , e gibi) sembolik gösterimlerini tercih etmeleri idi.

Ancak bilgisayar cebiri sistemleri matematiksel ifadelerde gösterdikleri bu başarıyı maalesef henüz grafik çıktılarında gösterebilmiş değildir. Schoenfeld (1992) bu konuyu ele aldığı çalışmasında, bunun öğrencilerin aklının karışması ya da yanlış kavrama gibi sonuçlarının olabileceğine dikkat çekmiştir. Kullanılan ekranın çözünürlük, boyut vb. özelliklerine bağımlılık, yanlış ölçeklendirme gibi birçok sebebe bağlanan bu problemlere örnek olarak Crowe ve Zand (2000), bir bilgisayar cebiri sisteminde aşağıdaki gibi sonuçlanan, biri diğerinin iki birim ötelenmiş olan iki parabolün çizimini vermiştir:

Şekil 16

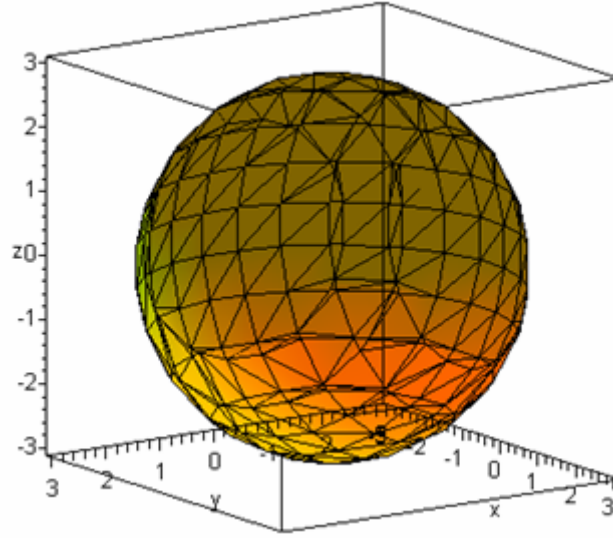
$f(x)$ ve $f(x)+2$ parabollemi.



$f(x)$ ve $f(x)+2$ parabollemi aralarındaki fark 2 birim olmasına karşın, çıktı grafikteki ölçek problemi sebebiyle hiç de öyle durmamaktadır.

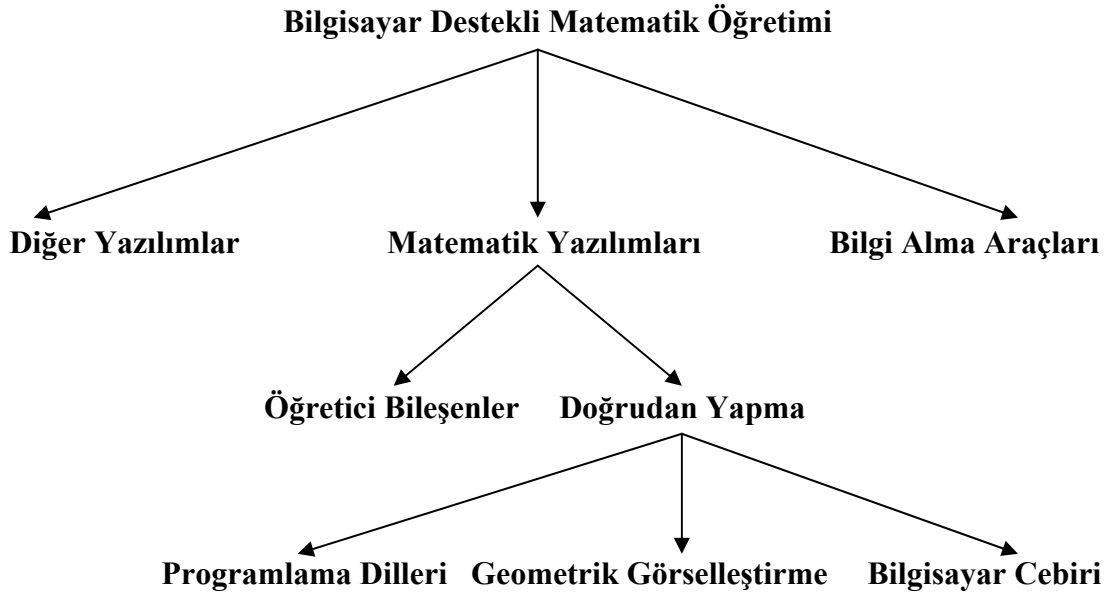
Son yıllarda özellikle üzerinde durulan bu problem, çoğu bilgisayar cebiri sistemi tarafından bir ölçüde de olsa çözülmüştür. Örneğin Maple’da, *plots* bileşeninde yer alan *implicitplot3d* fonksiyonu kullanılarak gerçekleştirilen küre çizimleri yazılımın dokuzuncu sürümüne kadar (ekranın dikdörtgensel yapısı temel alınarak ölçeklendirme yapıldığından) küreden çok bir elipsoide benzemektedir. Oysa yeni sürümlerde bu sorun giderilmiştir. Şekil 17’deki küre çizimi Maple10 kullanılarak oluşturulmuştur ve gerek oran, gerekse ölçeklendirme açısından oldukça iyi bir grafik çıktısı sayılabilir:

Şekil 17
Maple 10 da çizilmiş bir küre.



Crowe ve Zand, üniversite seviyesinde matematik eğitiminde bilgisayarın rolü ile ilgili olarak yayınladıkları seri makalelerinin ilkinde (2000), konuyla ilgili tartışma başlatmak amacıyla, bilgisayar destekli matematik öğretimi için matematik yazılımları, bilgi alma araçları ve diğer yazılımlardan oluşan bir sınıflandırma yapmışlardır. Bu sınıflandırmada yer alan matematik yazılımları için daha önceden Tall (1994) tarafından yapılan sınıflandırmayı da göz önüne alırsak, Şekil 18 deki gibi bir şema elde ederiz.

Şekil 18
Crowe ve Zand'in sınıflandırması.



“Diğer yazılımlar” ile, amacı matematiği öğretmek olmayan, daha çok genel amaçlı olarak geliştirilmiş yazılımların, matematik eğitiminde kullanılabilirliği anlatılmaktadır. Buna örnek olarak, Sutherland (1994)’ın, öğrencilere cebirin temellerini kavratmak için bir elektronik tablola yazılımı olan Excel’i kullanması gösterilebilir. Sutherland aynı çalışmasında, matematikteki “değişken” kavramını öğretmek için de Logo yazılımından faydalanmıştır, ancak bu yazılım daha çok, sınıflandırmadaki “matematik yazılımları” kategorisine girmektedir.

Elektronik tablola yazılımları, temel fonksiyon grafiklerinin öğretiminde de kullanılabilir. Bu öğretim, öğrenciye iki farklı açıdan yarar sağlayacaktır: Birincisi, öğrencinin temel grafikleri öğrenmesi için daha motive edici bir ortam sağlanacaktır, ikincisi ise, öğrencinin, “fonksiyon grafiği çizme” etkinliğinin yazılım tarafından en ilkel şekliyle, yani doğrudan belirli bir aralıkta, belirli bir adım sayısı ile değer alan x değerlerine karşılık gelen y değerlerinin fonksiyon kullanılarak hesaplanması, daha sonra bu değerlerin yazılımın grafik aracı kullanılarak grafiğe dönüştürülmesi şeklinde gerçekleştirildiğinin bilincinde olması, onun fonksiyon ve

grafîği arasındaki mantıksal ilişkiyi kurabilmesi açısından da yararlı olacaktır. Aynı işlem bir programlama dili kullanılarak da gerçekleştirilebilir, ancak bunun için öğrencinin en az bir programlama diline iyi seviyede hâkim olması beklenmelidir. Oysa yukarıda anlatılan etkinliğin gerçekleştirilebilmesi için elektronik tablola yazılımında temel seviyede kullanıcı bilgisine sahip olmak yeterli olacaktır.

Aşağıda, popüler elektronik tablola yazılımı olan Excel kullanılarak gerçekleştirilmiş bir etkinlik, adım adım verilmiştir. Etkinlikte, temel trigonometrik fonksiyonlardan sinüs fonksiyonunun grafikleri yazılımın “çizgi grafik” aracı kullanılarak çizdirilmektedir.

1. Adım: Fonksiyon için belirlenen tanım kümesini $(-360^\circ, 360^\circ)$ oluşturan x değerleri bir sütuna girilir (Şekil 19-A sütunu).

2. Adım: x değerleri, yazılımın sinüs fonksiyonunun kabul ettiği şekilde, radyan türüne dönüştürülür. Elde edilen yeni değerler, yeni bir veri sütunu olarak oluşturulur (Şekil 19-B sütunu).

3. Adım: Radyan değerlerinden oluşan sütundaki hücrelere sinüs fonksiyonu uygulanır ve elde edilen sonuçlar, üçüncü bir sütunu oluşturur (Şekil 19-C sütunu).

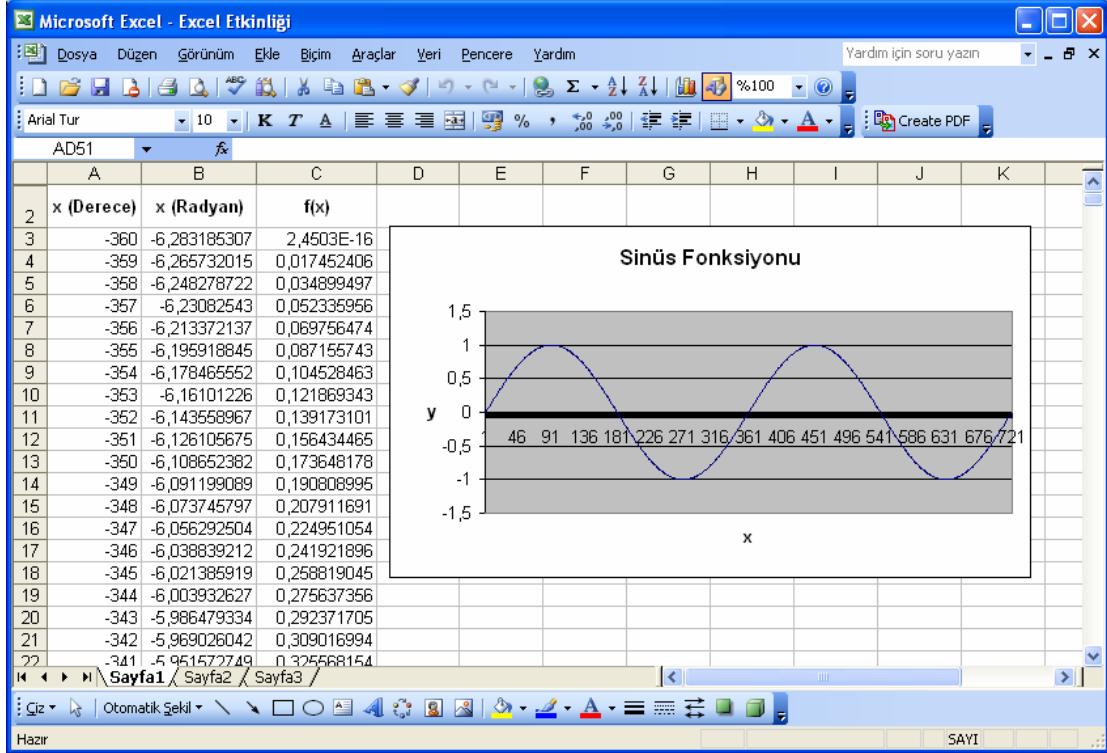
Şekil 19
Excel’de fonksiyon grafikleri.

	A	B	C	D	E	F	G	H	I	J
2	x (Derece)	x (Radyan)	f(x)							
3	-360	-6,283185307	2,4503E-16							
4	-359	-6,265732015	0,017452406							
5	-358	-6,248278722	0,034899497							
6	-357	-6,23082543	0,052335956							
7	-356	-6,213372137	0,069756474							
8	-355	-6,195918845	0,087155743							
9	-354	-6,178465552	0,104528463							
10	-353	-6,16101226	0,121869343							
11	-352	-6,143558967	0,139173101							
12	-351	-6,126105675	0,156434465							
13	-350	-6,108652382	0,173648178							
14	-349	-6,091199089	0,190808995							
15	-348	-6,073745797	0,207911691							
16	-347	-6,056292504	0,224951054							
17	-346	-6,038839212	0,241921896							
18	-345	-6,021385919	0,258819045							
19	-344	-6,003932627	0,275637356							

4. Adım: Elde edilen $f(x)$ değerleri (Şekil 19-C sütunu) kullanılarak, Excel’in grafik oluşturma aracında yer alan “çizgi” kategorisinde bir grafik oluşturulur (Şekil 20).

Bunların yanı sıra, amaçları matematiği sevdirmek ya da sadece hoş vakit geçirtmek bile olsa, her türlü eğitsel matematik oyun yazılımları da bu kategoriye sokulabilir. Eğitsel oyunların amacı, öğrenciye hoşça vakit geçirtmenin yanı sıra, onu hedeflenen düzeye getirmektir. Bu amaçla eğitsel oyunlar, yapay zekânın da yardımıyla, öğrenciyi oyun sırasında izleyerek, gelişimini değerlendirebilir ve oyunu öğrenciye özel olacak şekilde, eksikliği hissedilen konulara ağırlık vermek suretiyle yönlendirebilir.

Şekil 20
Excel’de fonksiyon grafikleri.



Bilgi alma araçları, büyük ölçüde internete dayanan teknolojileri kapsamaktadır. İçeriği matematik eğitimi olan siteler ve dağıtımı büyük ölçüde internet üzerinden yapılan elektronik kitaplar bu kapsam içinde yer alırlar.

Matematik yazılımlarını kullanım açısından iki farklı şekilde düşünebiliriz. Bunlar, yazılım kullanılarak geliştirilmiş ve yazılımın kullanımı üzerine çok fazla beceri gerektirmeyen “öğretici bileşenler” ve “doğrudan yapma” olarak adlandırdığımız, yazılımın daha profesyonel kullanımını kapsamaktadır. İlerleyen kesimlerde Maple için örnekleri verilecek olan öğretici bileşenler (“educational packages” ya da “tutors”), daha çok kullanıcı için hazırlanmış özel arayüzlerden oluşmaktadırlar. Kullanıcı-dostu olan bu arayüz öğrenciyi, yazılımın karışık sistemi ile meşgul olmaktan çok, konuya odaklanmaya teşvik edecektir. Böylece arka tarafta çalışmasını sürdüren yazılım bütün işlemleri gerçekleştirecek, sonuçlar ve girdiler bu özel arayüz kullanılarak yapılacaktır.

Diğer yandan, yazılımın doğrudan kullanımı şüphesiz ki kullanıcıya çok daha geniş bir ufuk sağlayacaktır. Ancak bunun için kullanıcının bazı konularda bilgi birikimi sahibi olması kaçınılmaz olacaktır. Bunlar, geometrik görselleştirmeler için iyi bir seviyede geometri bilgisi, sistemin girdi ve çıktı düzenine alışkın olmak, ya da daha ileri seviyede düşünülürse, yeni geliştirilen algoritmaları anlamak ve sisteme girmek açısından bilgisayar cebiri bilgisi ve bazı karmaşık işlemleri gerçekleştirecek prosedürleri yazabilmek için programlama deneyimi olarak sıralanabilir.

Yaptığı çalışmada Tall, matematik eğitimcilerini, herhangi bir bilgisayar cebiri sistemini kullanma becerisine sahip olmaları gerekmeyen kişiler olarak tanımlamış, bunun yerine, kendileri için hazırlanmış öğrenme ortamlarını kullanabilmelerinin yeterli olacağını vurgulamıştır. Tall'ın matematik yazılımlarında yaptığı “öğretici bileşenler” ve “doğrudan gerçekleştirme” şeklindeki kesin ayrıma rağmen Crowe ve Zand, bilgisayar cebiri sistemlerinin her yönüyle öğrenme ve öğretmede önemli bir rolü olduğuna inanmaktadırlar. Bu düşüncelerine destek olarak da, henüz kişisel bilgisayarların yaygınlaşmadığı ve bilgisayar cebiri konusundaki çalışmaların başlangıç aşamasında olduğu bir dönemde, Stoutmeyer (1979) tarafından yazılan ve bilgisayar cebiri sistemlerinin pedagojik açıdan yararlarını vurgulayan makalesini göstermektedirler. Stoutmeyer'in o devirde, kendi ifadesi ile “radikal bir öneri” olan düşüncesi, günümüzde gerçekten de matematik eğitiminde vazgeçilmez uygulamalar arasında yer almıştır.

Crowe ve Zand, makale serilerinin ikincisinde (2001), günümüzde gerek matematikçilerin, gerekse matematik eğitimcilerinin her zamankinden daha çok bilgisayarları kullanma eğiliminde olduklarına dikkat çekmektedirler. Buna sebep olarak da iki olguyu öne sürmektedirler. Bunlardan ilki, günümüzde bilgisayarların hemen hemen her konunun eğitiminde artan rolleri, ikincisi ise matematik eğitimcilerinin (ilköğretim seviyesindeki öğretmenlerden üniversite seviyesindeki öğretim üyelerine kadar) artık soyut kavramları somutlaştırmak için daha fazla “araca” sahip olma istekleri olarak vurgulanmaktadır. Bu konuda en önemli araçlardan birisi de kuşkusuz ki bilgisayar cebiri yazılımlarıdır. Bu sebeple,

yazılımların geliştirilmesinde çalışan gruplarda, matematikçilerin yanı sıra eğitim teknologları da yer almaya başlamışlardır. Çünkü artık, bu yazılımlar sadece birer “bilimsel programlama” aracı değil, aynı zamanda matematik eğitimcilerinin en önemli materyallerinden birisi haline gelmiş “eğitim materyalleri”dirler.

Crowe ve Zand aynı makalelerinde, elektronik tablola yazılımlarının matematik eğitimindeki kullanımlarını tekrar vurgulamakta, Golshan (1997) tarafından, üniversite seviyesindeki bazı analiz dersine giriş konularının öğretimi için Excel kullanılarak geliştirilen örneklere dikkat çekmektedirler. Golshan bu çalışmada, fonksiyonların maksimum ve minimumlarının belirlenmesi, denklemlerin yaklaşık köklerinin bulunması, denklem sistemlerinin yaklaşık çözümleri, polinomlar, olasılık problemleri ve Binom açılımları konularını kapsayan birçok örneği, Excel kullanarak başarıyla gerçekleştirmiştir. Benzer şekilde, ancak elektronik tablola yazılımlarının bilgisayar cebiri sistemleri ile kombine bir şekilde kullanıldığı bazı çalışmalar da olmuştur. Bunlara örnek olarak Shelton (1995)’un, Golshan tarafından çalışılan içeriğe benzer bir içeriği Excel ve Mathematica yazılımlarını birlikte kullanarak geliştirdiği çalışması ve Swartz (1996)’ın, Laurent serileri, rezidüler gibi “karmaşık analiz” konularını öğretmek için yazdığı Maple program kodlarının sonuçlarını Excel kullanarak gösterdiği çalışması verilebilir. Bu çalışmada Swartz, Excel’i bir bakıma Maple için arayüz olarak kullanmıştır. Günümüzde ise bilgisayar cebiri yazılımları, bu gereksinimi kendi içlerinde çözme yoluna gitmişler, sistemlerine “kullanıcı dostu” arayüz tasarımı için çeşitli araçlar eklemişlerdir.

4.3. Bilgisayar Destekli Matematik Öğretimi Materyali Örnekleri

Bu kesimde, bilgisayar destekli matematik öğretimi için Maple yazılımı kullanılarak hazırlanmış bazı materyal örnekleri tanıtılmıştır. Bu örnekler, Maple yazılımında, maplet yapısı kullanılarak geliştirilmiştir.

4.3.1. Maple’da Maplet Kavramı

Maple, 8 versiyonu ile geliştirdiği *Maplet* kavramı ile, arka alanda Maple motorunu kullanan kullanıcı ara yüzü (user interface) tasarlama imkanı tanımıştır. Script tarzında bir programlama dili kullanılarak geliştirilen uygulamalar, yazılımın dokuzuncu sürümüne kadar yazılım içerisinde başlatılıyorken, bu sürümde her bir maplet’in, yazılımdan ayrı bir şekilde program dışından da çalıştırılabilir olmasına olanak tanınmıştır. Ancak arka planda yapılan hesaplamalar yine yazılımın kendisi tarafından yapıldığından, yazılımın kendisinin de aynı bilgisayarda kurulu olması gerekmektedir. Böylelikle kullanıcılar, yazılımın güçlü hesaplama yeteneklerini, sistemin, iyi bir seviyede programlama dili mantığı bilmeyi gerektiren karmaşık kullanımını öğrenmeye gerek duymadan kullanabileceklerdir. Ayrıca yazılım da kendisi haricinde bir arayüz programına (Excel gibi) ihtiyaç duymayacaktır.

Bu teknoloji, özellikle bilgisayar cebiri sistemlerinin matematik eğitim alanında kullanılmalarını hızlandırmıştır. Çünkü böyle bir tasarımı kullanmak için, maplet geliştirme dilinin eğitimciler tarafından bilinmesi zorunlu değildir, profesyonel programcılar (ve doğal olarak matematikçiler) tarafından geliştirilen uygulamaları kullanacak öğretmenler, hemen hemen her konunun anlatımına bilgisayar desteği getirebilirler. Bu kullanıcı ara yüzlerinin öğrenciler tarafından da kolaylıkla kullanılabilir olmaları, matematik eğitimine önemli bir katkı sağlayacaktır.

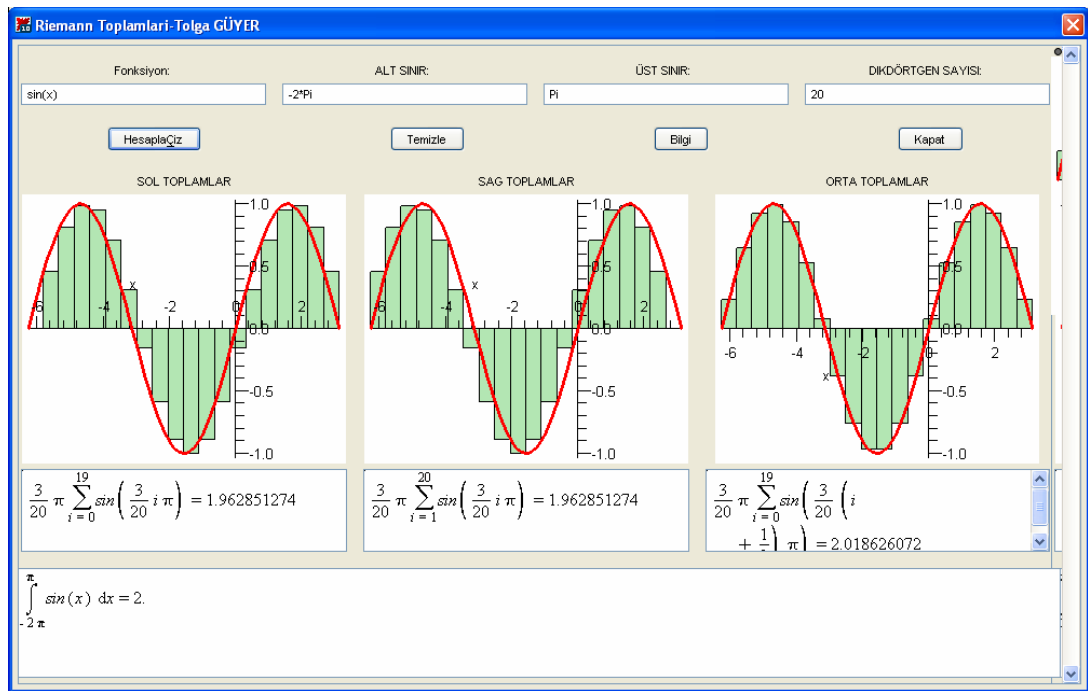
İlerleyen kesimlerde bazı konuların öğretimine yönelik olarak geliştirilen maplet örnekleri verilmiştir.

4.3.2. Riemann Toplamları Mapleti

İntegral kavramının anlaşılması için oldukça büyük bir öneme sahip olan Riemann toplamları, benzetim (simulasyon) tekniğine çok uygun bir konudur. Aşağıda ekran görüntüsü verilen maplet, verilen fonksiyon ve aralık için sağ, sol ve orta Riemann toplamlarını, istenilen dikdörtgen sayısına göre hem grafikte

göstermekte, hem toplam formülünü bulmakta, hem de sayısal sonucu hesaplamaktadır. Kullanıcı, aynı fonksiyon için dikdörtgen sayısını istediği gibi artırarak eğri altında kalan alana nasıl daha da yaklaşıldığını görsel ve sayısal olarak gözlemleyebilmektedir.

Şekil 21
Riemann toplamları mapleti.



4.3.3. Adım Adım Öğreticiler

Adım adım öğreticiler, Maple yazılım grubu tarafından geliştirilmiş ve yazılımın uygulamalar sitesinde (<http://www.mapleapps.com>), eğitim kategorisi altında kullanıcılara ücretsiz olarak sunulmaktadır. Bu uygulamalar biraz da, genel olarak bilgisayar cebiri sistemlerine karşı “kapalı kutu” olmalarından kaynaklanan önyargılı yaklaşımı bir ölçüde de olsa azaltmıştır. Amaçları sadece sonuç almak olan matematikçiler ve mühendisler için bu çok önemli olmasa da, işin içerisinde eğitim

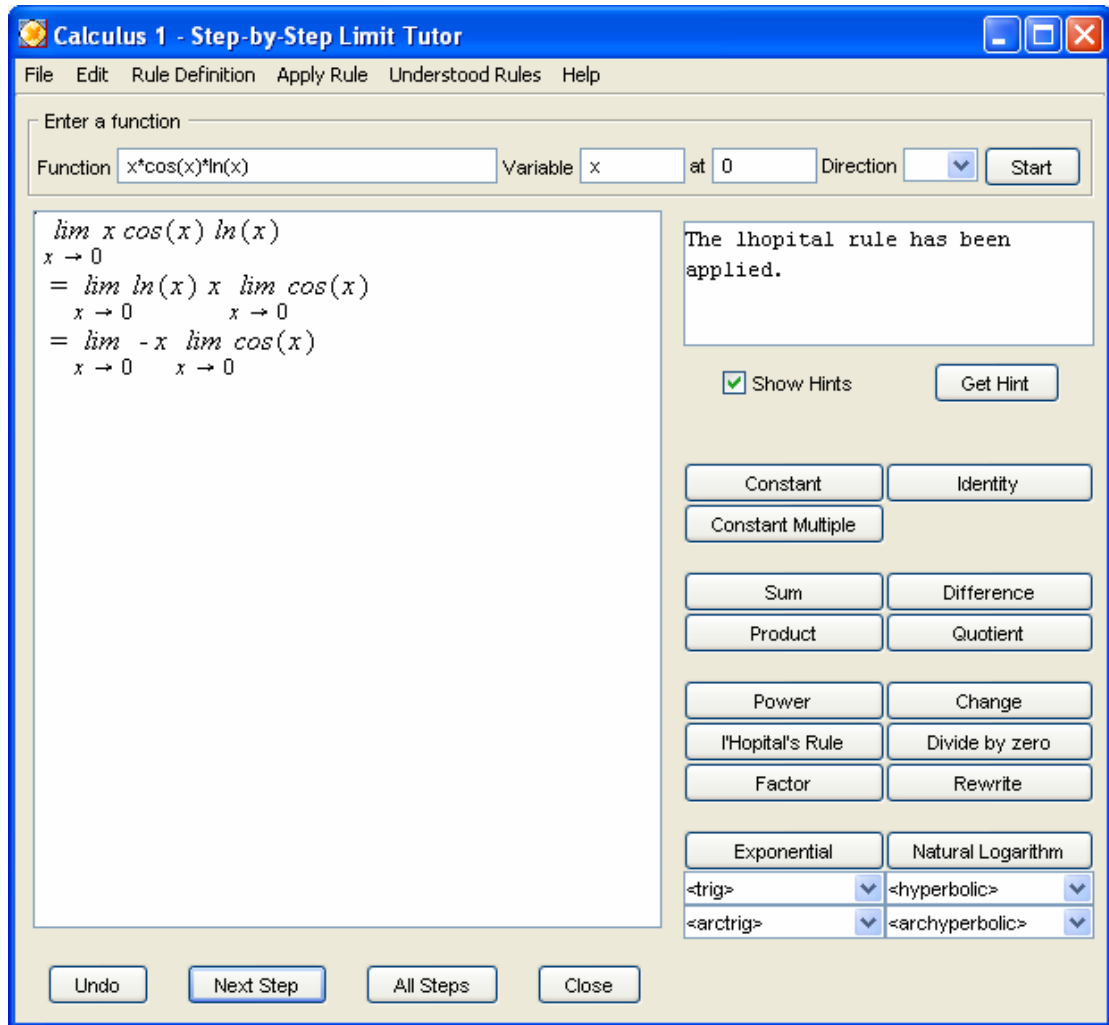
girdiğinde durum biraz farklılaşmaktadır. Çünkü artık yazılımın sonucu ne olarak bulunduğu kadar, nasıl bulunduğu da önem kazanmaktadır.

Bilgisayar cebiri teorisyenleri için uzun süredir bilinen, ancak çözümü hiç de kolay olmayan bu problem, geliştirilen bu uygulamalarla kısmen de olsa çözülmüş gözükmektedir. Çünkü teorisyenler için son derece anlamlı olan çözüm adımları, matematiği öğrenen konumda olan bir kullanıcının da ötesinde, doğrudan bilgisayar cebiri ile ilgilenmeyen bir matematikçi için de anlaması kolay olmayan adımlardır. Dolayısıyla çözümdeki her bir adımın, matematiği öğrenen bir öğrencinin anlayabileceği dile dönüştürülmesi işlemi yine bilgisayar cebiri teorisyenlerinin işi olmuştur.

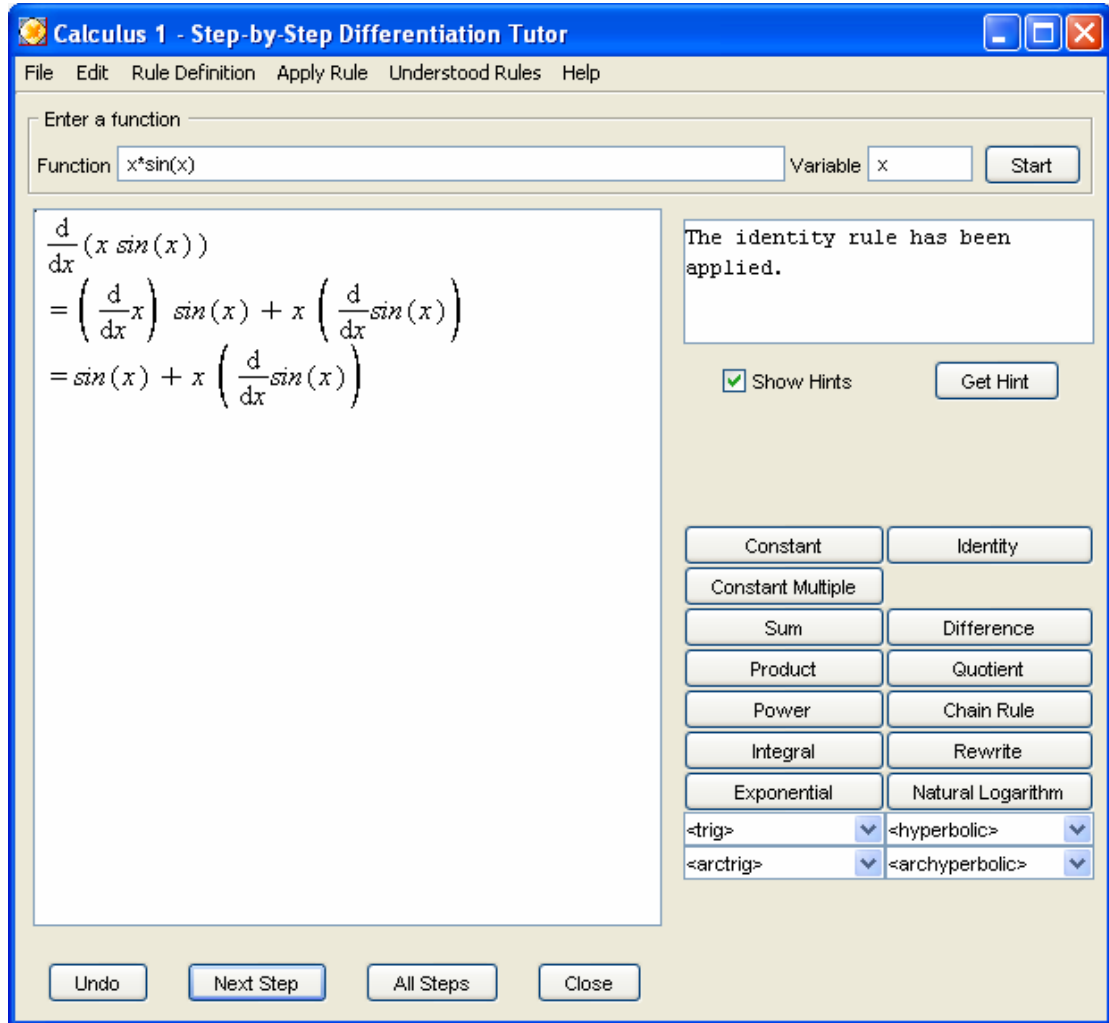
Aşağıdaki kesimlerde, analizin üç temel konusu olan limit, türev ve integral konuları için geliştirilmiş üç farklı öğretici maplete ait ekran görüntüleri verilmiştir. Bu uygulamaların her birinde, fonksiyon ve uygulama için gereken diğer veriler (nokta ya da aralık gibi) kullanıcı tarafından girilmekte, kullanıcı isterse çözümü adım adım izleyebilmekte, istediği takdirde ise her bir aşama için uygulamaya uygun olarak çözüm önerisinde bulunabilmektedir. Örneğin integral alma uygulamasında, değişken değiştirme veya kısmi integrasyon gibi yöntem önerisinde bulunabilmektedir. Bunun karşılığında, önerisinin doğruluğu hakkında sistemden geri bildirim de alabilmektedir.

Bu şekliyle, aşağıda örnekleri verilen uygulamalarda olduğu gibi, belirli kurallar dâhilinde gerçekleştirilen sembolik hesaplama işlemleri için geliştirilecek mapletler, konunun teorik boyutunun öğretilmesinde çok fazla katkı sağlamamakla birlikte, işlemlerde pratiklik kazandırma açısından oldukça önemlidir.

Şekil 22
Adım adım “Limit Hesaplama” öğreticisi.



Şekil 23
Adım adım “Türev Alma” öğreticisi.



Şekil 24
Adım adım “İntegral Alma” öğreticisi.

Calculus 1 - Step-by-Step Integration Tutor

File Edit Rule Definition Apply Rule Understood Rules Help

Enter a function

Function Variable from to

$$\int \sin^2 x \, dx$$

$$= \int \left(\frac{1}{2} - \frac{1}{2} \cos(2x) \right) dx$$

$$= \int \frac{1}{2} dx + \int -\frac{1}{2} \cos(2x) dx$$

$$= \frac{1}{2} x + \int -\frac{1}{2} \cos(2x) dx$$

The constant rule has been applied.

☒ Show Hints

Constant	Identity
Constant Multiple	
Sum	Difference
Power	Parts
Change	Revert
Solve	Rewrite
Exponential	Natural Logarithm
<trig>	<hyperbolic>
<arctrig>	<archyperbolic>

BÖLÜM 5

SONUÇ VE ÖNERİLER

Gelişen donanım ve yazılım teknolojisine bağlı olarak yetenekleri her geçen gün daha da artan bilgisayar cebiri sistemlerinin, kullanım alanları da giderek genişlemektedir. Bu yazılımları ve ek bileşenlerini geliştiren proje gruplarına son zamanlarda yeni bir meslek grubu da eklenmiştir; eğitimciler. Bu da göstermektedir ki, bu yazılımların gelecekte her seviyedeki matematik eğitiminde yoğun olarak yer almaları kaçınılmaz olacaktır.

Bilindiği üzere, matematik dersi özellikle soyut kavramların çokluğu nedeniyle öğrencilerin anlamakta güçlük çektiği bir alandır. Bu bağlamda, bilgisayar yazılımlarının matematik öğretiminde rollerini vurgulayan araştırmalar incelendiğinde açıkça görülmektedir ki; bilgisayar cebiri sistemleri özellikle soyut kavramların görsellik kazandırılarak somutlaştırılması ve anlaşılabilirliklerinin artırılmasında işlevsel bir öğretim materyalidir.

Dikkat çeken önemli bir nokta da, ülkemizde bu konuda yapılan teorik ve deneysel çalışmaların yetersizliğidir. Kuşkusuz bunun başlıca sebebi, tek başına matematik eğitimcilerinin bu yazılımları ileri düzeyde kullanma becerilerine sahip olmamalarıdır, ki bu da onlardan beklenmemelidir. Sonuç olarak, bu konuda yapılacak çalışmaların üç ayaklı bir dayanağa gereksinimi vardır. Bunlardan ilki, programlama becerisi, ikincisi, konuyla ilgili matematik bilgisi ve son olarak da alana özgü eğitim temelleri. Dolayısıyla, bu konuda yapılacak çalışmalar, bu alanların işbirliği ile gerçekleştirildiği takdirde, önemli sonuçlar elde edilebilecektir.

Diğer yandan, bilgisayar cebiri sistemleri kullanılarak geliştirilen matematik eğitimi materyallerinin, hemen hemen her okulda kurulu en az bir bilgisayar laboratuvarının var olduğu düşünülürse, kullanılabilecekleri uygun ortam problemleri olmayacaktır. Ayrıca bu yazılımlar, her konu veya konu grubu için ayrı ayrı en baştan geliştirilecek özel uygulamaların geliştirme süreçleri düşünüldüğünde, çok daha hızlı ve pratik çözümler sunacaklardır.

KAYNAKLAR

- Alkan, C., (1995), **Eğitim Teknolojisi**, Atilla Kitabevi, Ankara.
- Akkoyunlu, B., (1995), **Bilgi Teknolojilerinin Okullarda Kullanımı ve Öğretmenlerin Rolü**, Hacettepe Üniversitesi Eğitim Fakültesi Dergisi, 11, ss105-109.
- Akpınar, Y., (1995), **Bilgisayar Destekli Öğretim ve Uygulamalar**, Anı Yayıncılık.
- Alessi, S.M., Trollip, S.R., (1991), **Multimedia for Learning: Methods and Development**, Allyn and Bacon, USA.
- Batut, C., Bernardi, D., Cohen, H. and Olivier, M., (1995), **User's Guide to PARI-GP, version 1.39**, Université Bordeaux I.
- Butler, G., Cannon, J., (1990), **The Design of Cayley-A Language for Modern Algebra, Design and Implementation of Symbolic Computation Systems**, Lecture Notes in Computer Science, 42, Springer-Verlag.
- Capani, A., Niesi, G., (1995), **CoCoA User's Manual**, University of Genova.
- Carr, D.A., (1996), Toward More Understandable User Interface Specifications, **Design, Specification and Verification of Interactive Systems '96**.
- Cheb-Terrab, E.S., Duarte, L.G.S. and da Mota, L.A.C.P., (1997), Computer Algebra Solving of Second Order ODEs Using Symmetry Methods, submitted to **Computer Physics Communications**.
- Crowe, D., Zand, H., (2000), Computers and undergraduate mathematics I: setting the scene, **Computers&Education**, 35, 95-121.
- Crowe, D., Zand, H., (2001), Computers and undergraduate mathematics 2: on the desktop, **Computers&Education**, 37, 317-344.

- Davenport, J.H., Siret and Y., Tournier, E., (1993), **Computer Algebra, Systems and Algorithms for Algebraic Computation**, Academic Press.
- Dörfler, W., (1993), Computer use and views of the mind, In **Learning from computers: mathematics education and technology**, C. Keitel, K. Ruthven, Berlin: Springer-Verlag.
- Dreyfus, T., (1994), The role of cognitive tools in mathematics education, In **Didactics of mathematics as a scientific discipline**, R. Biehler, R.W. Scholz, R. Strasser, B.Winkelmann, Dordrecht: Kluwer.
- Fidan, N., Erden, M., (1983), Eğitime Giriş, Alkım Kitabevi Eğitim Kitapları.
- Golshan, B., (1997), Mathematics with spreadsheets, P. Bogacki (Ed.), **Electronic Proceedings of the Tenth Annual Conference on Technology in Collegiate Mathematics**, Chicago.
- Heck, A., (1996), **Introduction to Maple**, Springer-Verlag.
- Heinich, Robert., Molenda, Michael., & Russell, James D. (1993). **Instructional Media and the New Technologies of Instruction** - 4th edition, New York: Macmillan Publishing Company.
- Hollinger, C., Serf, P., (1991), **SIMATH - A Computer Algebra System, Computational Number Theory**, de Gruyter.
- Horwitz, P., and Feurzeig, W. (1994), Computer-aided inquiry in mathematics education, **The Journal of Computers in Mathematics and Science Teaching**, 13(3), 265.
- Joseph, L.G., Deborah, H. and Edward, J.S., (1999), User-Centered Design and Evaluation of virtual Environments, **IEEE Computer Graphics and Applications**, November, pp.51-59.
- Kajler, N., Soiffer, N., (1994), Some human interaction issues in computer algebra, **SIGCHI Bulletin**, 26(4), 64.

- Kayabaşı, Y., (2005), Sanal Gerçeklik ve Eğitim Amaçlı Kullanılması, **The Turkish Online Journal of Educational Technology**, Volume 4, Issue 3, Article 20.
- Lee, J.A.N., (1995), **Computer Pioneers**, IEEE Computer Society, Los Alamitos, CA, pp.51-64.
- Leeuwen, van M.A., Cohen, A.M. and Lisser, B., (1992), **LiE : A Package for Lie Group Computations**, CAN Expertise Center.
- Mitic, P., Thomas, P.G., (1994), Pitfalls and limitations of computer algebra, **Computers and Education**, 22(4), 355.
- Özdener, N., (2005), Deneysel Öğretim Yöntemlerinde Benzetişim (Simulation) Kullanımı, **The Turkish Online Journal of Educational Technology – TOJET**, volume 4 Issue 4 Article 13.
- Özdener, N., Erdoğan, B. (2001), Bilgisayar destekli eğitimde kullanım amaçlı bir simülasyonun tasarlanması ve geliştirilmesi, **Fen Bilimleri Eğitimi Sempozyumu ve Fuarı**, 7-9 Eylül, Maltepe Üniversitesi, İstanbul.
- Rose, F., (1989), **West Of Eden: The End Of Innocence At Apple Computer**, New York, Penguin Books, 33.
- Schmettow, G.v.J., (1991), **KANT- a Tool for Computations in Algebraic Number Fields, Computational Number Theory**, de Gruyter.
- Schneiderman, B., (1992), **Designing the user interface: Strategies for effective human-computer interaction**, Addison-Wesley Pub.
- Schoenert, M., et al, (1994), **GAP: Groups, Algorithms and Programming**, RWTH Aachen.
- Schoenfeld, A.H., On calculus and computers: thoughts about echnologically based calculus curricula that might make sense. In **Symbolic computation in undergraduate mathematics education**, Z.A. Karian, (MAA Notes 24).

- Shelton, T., (1995), Reducing tedium in teaching and learning, In: P. Bogacki (Ed.), **Electronic Proceedings of the Eighth Annual Conference on Technology in Collegiate Mathematics**, Chicago.
- Stillman, M., Bayer, D., (1989), **Macaulay User Manual**.
- Stoutmeyer, D. R., (1979), Computer symbolic math and education: a radical proposal, **Bulletin of the Special Interest Group in Symbolic Algebraic Manipulations of the Association of Computing Machinery**, 13(2), 8-24.
- Sutherland, R., (1994), The role of programming towards experimental mathematics. In **Didactics of mathematics as a scientific discipline**, R. Biehler, R. W. Scholz, R. Strasser, B. Winkelmann, Dordrecht: Kluwer
- Swartz, G. B., (1996), Complex variables using Excel and Maple, In: P. Bogacki (Ed.), **Electronic Proceedings of the Ninth Annual Conference on Technology in Collegiate Mathematics**, Chicago.
- Tall, D., (1994), Computer environments for the learning of mathematics. In **Didactics of mathematics as a scientific discipline**, R. Biehler, R. W. Scholz, R. Strasser, B. Winkelmann, Dordrecht: Kluwer
- Weik, M.H., (1961), **The ENIAC Story**, <<http://ftp.arl.mil/~mike/comphist/eniac-story.html>>.
- Williams, G. and Rob M., (1984), **The Apple Story, Part 1: Early History**, Byte Dec., A69.
- Williams, Michael R., (1997), **A History of Computing Technology**, (Second Edition), IEEE Press, Los Alamitos, CA, pp.154-186.
- Yalın, H. İ., (2001), **Öğretim Teknolojileri ve Materyal Geliştirme**, Nobel Yayınları.
- < <http://www.mathcad.com>>, Mathcad formal web site, (08 Haziran 2006).

<<http://www.gap-system.org/>>, GAP formal web site, (21 Nisan 2006).

<<http://www.maple.com>>, Maple formal web site, (11 Mayıs 2006).

<<http://www.maplesoft.com>>, Maple Software, (11 Mayıs 2006).

<<http://www.stephenwolfram.com>>, Mathematica formal web site, (10 Mart 2006).

<<http://www.thocp.net>>, (07 Nisan 2006).

<http://www.maplesoft.com/applications/app_center_browse.aspx?CID=14>, Maple Application Centre – Education Powertools, (09 Mayıs 2006).