

Temel Bilimler ve Mühendislik Eğitiminde

Programlama Dili Olarak Python

Berkin Malkoç

İstanbul Teknik Üniversitesi, Fizik Mühendisliği Bölümü, İstanbul
malkocb@itu.edu.tr

Özet: Mühendislik ve temel bilimler lisans öğrenimindeki 'hesaplamalı bilim eğitimi'nde programlama dili olarak Python'ın kullanılması, mevcut koşullar ve sağlayacağı avantajlar açısından doğru bir tercih olacaktır gibi görünmektedir. Python, öğrencilerin hem akademik hayatlarındaki bilgisayarlı çalışmalarda; hem de ileride bilişim ile ilgili yapabilecekleri diğer işlerde daha iyi bir başlangıç yapabilmelerine yardımcı olacaktır. Bu makalede, hesaplamalı bilim eğitiminde Python'ın kullanılması önerisi gerekçelendirildikten sonra, Python'a dayalı ders yürütecek eğitimcilere bu dersleri için yaklaşım ve kaynak önerileri özetlenmektedir.

Anahtar Sözcükler: Sayısal ve Sembolik Hesaplama, Bilimsel Hesaplama, Bilim, Mühendislik, Lisans Öğretimi, Programlama Dili, Python

Python as the Programming Language for Undergraduate Education in Basic Sciences and Engineering

Abstract: The use of Python as the programming language for the computational science courses in engineering and basic sciences curricula seems to be the optimal choice, considering the needs and the constraints of the programs and the advantages that Python brings. Python would enable the students to make a quick and satisfactory start both in their ability to use computers for sciences/engineering needs during their education and in various contexts during their professional careers. In this article, the argument for the use of Python in computational science and engineering education will be detailed and suggestions and educational resources for the educators adopting Python will be summarized.

Keywords: Numerical and Symbolic Computing, Scientific Computation, Science, Engineering, Undergraduate Training, Programming Language, Python

1. Giriş

1.1. Python

Python, ilk sürümü Guido van Rossum tarafından 1991'de ortaya konulmuş genel amaçlı bir programlama dilidir. Yorumlanan ve dinamik bir dil olan Python, esas olarak prosedürel ve nesne tabanlı programlama yaklaşımlarını ve belli bir oranda da fonksiyonel programlamayı desteklemektedir. Python Yazılım Vakfı¹, Python'ın ana gerçekleştirimi olan C dili ger-

çekleşimini² özgür ve açık kaynak kod³ mantığı altında yürütmekte ve Python'ın fikrinsel haklarını korumaktadır. Günümüzde Python, görece kolaylığı ve sahip olduğu geniş standart kütüphane sayesinde oldukça popülerleşmiş ve büyük kurumların da arasında olduğu yaygın bir kullanıcı kitlesine ulaşmıştır.

1 Python Software Foundation

2 CPython

3 Kısaca "Ö/AKK," olarak ifade edeceğiz.

1.2. Bilimsel Hesaplama Python Kullanımı⁴

Python, 2000’li yıllardan itibaren bilimsel veya mühendislikle ilgili hesaplamalı çalışmalarda da çokça kullanılmaya başlamıştır. Bunda hem donanımsal hem de yazılımsal gelişmelerin etkisi olmuştur: Donanımsal açıdan, işlem gücü ve bellek/depolama kapasitesinin herkesin kolayca ulaşabileceği şekilde artıp yaygınlaşması ile, Python gibi yorumlanan dillerin bazı işlemlerde epeyce yavaş olmaları ve yüksek düzeyli veri yapılarının kullanılmasının hafızada çok yer tutması gibi olumsuzluklar birçok durumda göz ardı edilebilecek seviyeye gerilemiştir. Yazılımsal açıdan ise, İnternet ve onunla beraber Ö/AKK akımının yaygınlık kazanması, bu yaklaşımı benimseyen bilimcilerin İnternet üzerinden ve geliştirilen verimli araçlar sayesinde eşgüdümlü bir şekilde çalışmalarını beraberinde getirmiştir.

Sonuç olarak, günümüzde, www.scipy.org etrafında örgütlenmiş bir geliştirici topluluğu, bilimsel hesaplamaların temel ihtiyaçları ve özgül diğer uygulamalar için çeşitli Python kütüphanelerini olgun bir noktaya getirmiştir. Bu camianın iletişimi, İnternet dışında, her sene üç farklı kıtada düzenlenen ve yüzler mertebesinde kişinin katılımıyla gerçekleşen “bilimde Python” konferanslarıyla sağlanmaktadır (ABD’de ‘SciPy’, Avrupa’da ‘EuroSciPy’ ve Hindistan’da ‘SciPy India’).

Python ile temel bilimsel hesaplama için üç ana kütüphane kullanılmaktadır: Hızlı dizi yapıları ve matris işlemleri gibi bazı temel işlevler için Numpy; Numpy veri yapılarının üzerinde sayısal entegrasyon, diferansiyel denklem çözümü, optimizasyon ve istatistik gibi herkesçe ihtiyaç duyulabilecek işlevler için SciPy ve iki boyutlu ve belli bir düzeyde üç boyutlu görselleştirme için Matplotlib (Tablo 1). Yukarıda sayılan kütüphanelerin dışında, bilimsel çalışmalarda sıkça yapılan veritabanı ve İnternet sayfalarının manipülasyonu gibi işlemler için gelişmiş ve

kararlı standart kütüphane bileşenleri vardır. Ayrıca belli bilim dallarına özgül, örneğin biyoloji için BioPython [1] gibi, kapsamlı kütüphaneler geliştirilmiştir. Özgül çalışma alanları için mevcut kütüphaneler konusunda okuyucu SciPy gibi sitelere başvurabilir. Bunların dışında bilimsel hesaplamada çokça kullanılan Fortran, C, R gibi dillerin Python içinden çağrılabilmesi için pratik arayüzler, GNU bilimsel kütüphanesi (‘GNU scientific library’) gibi kütüphaneler için bağlama kütüphaneleri ve paralel hesaplamalar için çeşitli kütüphaneler mevcuttur (bkz. Tablo 2). Python ile bilimsel hesaplama yaparken kullanılacak geliştirme ortamları seçenekleri de oldukça zengindir: Öncelikle, elbette, IDLE veya Eclipse gibi genel (Python) geliştirme araçları kullanılabilir. Bilimsel geliştiriciler için bunlardan daha pratik olarak MATLAB ortamına çok benzer bir ortam sağlayan üç farklı seçenek vardır (bkz. Tablo 3). Bu tür ortamların, MATLAB kullanıcılarının aşına olduğu ve değişkenleri yerinde değiştirme gibi olanaklar sağlayan “variable explorer” ve dili yeni öğrenmekte olanlar için çok faydalı olabilecek “object explorer”, kod analizcisi “Pylint” gibi özellikler vardır. Mathematica veya Maple kullanıcılarının aşına olduğu defter (‘notebook’) arayüzünü elde etmek içinse, Sage⁵ veya IPython (sürüm > 0.12) kullanılabilir. (IPython, bilimsel hesaplama özgü olarak geliştirilmiş ve standart olarak kullanılan yetenekli bir komut satırı Python ortamıdır. Diğer birçok proje (örn. Spyder), zengin etkileşimli programlama becerilerini IPython üzerinden sağlamaktadırlar.)

Yukarıda Python’ın çok yaygın kabul gördüğü bilimsel hesaplama alanındaki kullanımına kısaca değindik. Bildirinin geri kalan kısmında, üniversitelerimizin hemen hepsinde yürütülmekte olan bilimsel hesaplama eğitimi için Python’ın iyi bir programlama dili seçimi olduğunu, diğer seçeneklerle karşılaştırma da yaparak, göstermeye çalışacağız. Akabinde, Python’ı tercih edecek öğretim elemanlarının kullanabilecekleri kaynakları ve derslerde verimi arttıracak bazı uygulamaları ele alacağız.

4 Temel bilim ve mühendislik gibi çeşitli dallardaki problemlerin çözümleri için bilgisayarların kullanılmasını bu bildiride kısaca “bilimsel hesaplama” olarak adlandıracamız.

5

Sage’i bir sonraki kısımda ayrıca ele alacağız.

2. Bilimsel Hesaplama Eğitiminde Python

2.1. Python'un Olumlu Yönleri

Bu alt kısımda, Python'ın bilimsel hesaplama eğitimi açısından olumlu yönleri özetlenecek; bir sonraki altkısımda ise, burada değinilen nitelikler üzerinden hesaplamalı bilim eğitimi de kullanılabilir diğer alternatiflerle bir karşılaştırma sunulacaktır.

Özgür olması Python programlama dili, standart sürümünde, C dili kullanılarak gerçekleştirilmiştir. Bu kodlar ile Python'ın standart kütüphanesi, geliştirme araçları ve diğer birçok kütüphane, Ö/AKK olarak İnternet'ten ücretsiz ve herhangi bir lisans sorunu yaşanmadan indirilebilmektedir [2]. Dolayısıyla bilimsel hesaplama derslerindeki tüm öğrenciler istedikleri her yerde bu araçları serbestçe kullanabileceklerdir. Ayrıca, Python'un bilimsel kütüphaneleri de açık kaynak kodlu olduklarından, derste ele alınan konular için kullandıkları modüllerin kodlarını inceleyebilecek ve gereği halinde değiştirerek farklı amaçlara uyarlayabileceklerdir.

Kolay olması Python, kolay öğrenilebilirlik ve kullanım için tasarlanmıştır [3]. Bu kolay öğrenilebilirlik özelliği, doğal olarak, her türlü kullanım için önemli bir avantajdır. Bilimsel hesaplama bağlamında ise, gerektiğinde bilimsel hesaplama derslerinde dilin en baştan öğrencilere öğretilebilmesi ve yine de asıl işlemek istenilen konulara zaman ayrılabilmesi olasılığını ortaya koymaktadır. Şayet öğrenciler bilimsel hesaplama kapsamındaki ilk derslerine gelmeden önceki programlama derslerini de bu dilde alabilirlerse, daha çok konuyu daha derinlemesine işleme olanağı olacaktır.

Her ahvalde, Python'ın basitliği, öğrenilmek istenilen algoritmalara veya bilimsel problemlere odaklanılmasını kolaylaştıracaktır. Ayrıca öğrenciler hem kendi yazdıkları hem de başkalarının (özellikle de hocalarının) yazmış olduğu kodu daha kolay okuyabilecek ve kodda yatan fikri daha kolay özümseyebileceklerdir.

Etkileşimli olması Yorumlanan bir dil olması itibarıyla Python, yorumlayıcı üzerinde etkileşimli olarak çalışılmasına olanak sağlamaktadır. Bu özelliğin getirdiği üç önemli fayda vardır. Birincisi, dilin yeni özelliklerinin veyahut yeni kütüphanelerin öğrenilmesi sırasında hızlı ve etkileşimli bir şekilde 'keşif' yapmaya olanak sağlamasıdır. İkincisi, yeni bir program yazılırken hızlı bir şekilde deneme yapılmasının kolaylaşmasıdır. Bilimsel hesaplama özgü olan üçüncü fayda ise, hesaplamaların sonuçlarını adım adım görmeye ve eldeki problemin şekillenmesini verimli bir şekilde takip etmeye yardımcı olmasıdır.

Nesne yönelimli programlamayı desteklemesi Python'ın desteklediği temel paradigmlar prosedürel, nesne yönelimli ve fonksiyonel programlama paradigmlarıdır [4]. Bilimsel hesaplama prosedürel olarak kolayca çalışabilmek temel bir gerekliliktir. Görece büyük bilimsel yazılım projelerinde ise sağladığı kolaylık açısından nesne yönelimli programlamayı destekleyen bir dilin kullanımı tercih edilir. Ayrıca bu paradigma, özellikle de fiziksel benzetim yapmak gibi amaçları olan çalışmalarda doğal bir çalışma çerçevesi oluşturur [5, 6].

Popüler olması Bir programlama dilinin popülerliğinin tespiti için standart bir yöntem olmamakla beraber, çeşitli yaklaşımlarla yapılmış ölçümler birbirleriyle hemen hemen aynı çıkmakta [7] ve Python'ın en popüler diller arasında olduğunu göstermektedirler. TIOBE sitesinin rakamlarıyla, Python, genel diller sıralamasında altıncı; betik dilleri sıralamasında ise standart İnternet dili olan PHP'nin arkasından ikinci sıradadır [8]. Ayrıca Google gibi firmaların da Python kullanmaya başlamış olmaları [9], dilin arkasına geniş bir kitleyi almasını sağlamaktadır.

Popülerliğin bir programlama dili ve onu öğrenenler için çeşitli getirileri vardır. Birincisi, şu anda çok popüler olan bir dil en azından yıllar mertebesinde süreler için geliştirilmeye ve genişletilmeye devam edecektir. Dolayısıyla,

Python gibi popüler bir dili öğrenen bir bilimsel hesaplama öğrencisi daha uzun bir süre zengin ve bakımlı bir araç setine sahip olacağını varsayabilir. Bunun yanında, popüler bir dilin bilgisine haiz olmak ve o dilde geliştirme yapmakta tecrübeli olmak iş olanakları açısından da avantaj getirir.

Bilimci topluluğunun, kütüphanelerinin ve araçlarının geniş olması Python, genel popülerliğinin yanında, yukarıda da belirtildiği gibi, bilimsel camiada da geniş bir kabul görmüş durumdadır. Daha da önemlisi, bu camia İnternet ve konferanslar üzerinden iyi bir iletişim ağı içindedir. Bilimsel kütüphaneler ve araçlar açısından Python’ın ne kadar zengin olduğunu ise Kısım 1.2 içerisinde özetlemiştik. Bütün bunlar göz önüne alındığında, Python ekosisteminin, bilimsel hesaplama öğrencisinin öğrenme sürecini, takıldığı yerlerde yardım bulabilmesini ve ileriki safhalarda projelere bizzat katkıda bulunup kendisini geliştirmesini kolaylaştıran bir nitelikte olduğu ortaya çıkmaktadır.

Sage Sage, çok büyük hedefleri olan Ö/AKK bir bilimsel ve matematiksel hesaplama ortamı geliştirme projesidir [10]. Sloganı “Magma, Maple, Mathematica ve MATLAB’e sürdürülebilir Ö/AKK bir alternatif üretmek” olan Sage, ilk sürümü 2005 yılında çıkarılmış olmasına rağmen halihazırda çok yetenekli bir düzeye gelmiştir. Bunun en temel sebebi, Sage’in, önceden mevcut PARI, Maxima gibi Ö/AKK bilimsel yazılımları tek bir çatı altında toplayarak geçmişin birikiminden faydalanma yoluna gitmiş olmasıdır.

Bu büyük proje kendisine dil olarak Python’ı seçmiştir. Bilimsel hesaplamanın hemen hemen her yönünü kapsayan veya kapsamayı hedefleyen bu projenin kullandığı dili öğrenmek, hele de henüz bir “dil tercihi” yapmamış olan lisans öğrencileri için, oldukça verimli bir tercih olacaktır. Ayrıca dil olarak Python yerine başka bir dili öğrenen kişi, Sage’i kullanmak istediğinde hangi özelliklerinin Python’dan kaynaklandığını hangilerinin altta yatan hazır

sistemden kaynaklandığını anlayamamak gibi zorluklar çekebilmektedir.⁶

Öğrenme kaynaklarının genişliği Her dil için olduğu gibi Python’ı öğrenmek için de çeşitli türden kaynaklar göz önünde bulundurulabilir:

- Resmi olarak sağlanan dokümantasyon ve öğretici belge⁷. Bu belgeler oldukça kapsamlıdır ve hem dili hem de standart kütüphaneyi öğrenmek isteyen kişiye derli toplu bir kaynak sağlarlar. Ayrıca Python, üçüncü kişilerin sağladığı kütüphaneler için de kullanılması teşvik edilen ve kolay kullanımlı bir dokümantasyon sistemine sahiptir.
- Bilgisayar bilimleri için Python’ı ele alan kaynaklar. Bilgisayar bilimleri öğrencileri için yazılmış ve programlamaya Python ile giriş yapmayı anlatan çok sayıda kitap, ders notu ve video mevcuttur.
- Doğa bilimleri ve mühendislikler için Python’ı ele alan kaynaklar. Bilim camiasındaki kullanımına paralel olarak, Python’ı bilimsel hesaplama çerçevesinde anlatan yine farklı formatlarda çok sayıda kaynak mevcuttur.

Eğitim kaynakları ilgili kısımda geniş olarak ele alınacaktır (Kısım 4). Belki burada vurgulanması gereken nokta, bu kaynakların önemli bir bölümünün ücretsiz ve serbest olarak ulaşılabilir olmasıdır.

Farklı dillerle genişletilebilmesi **Yukarıda bahsi geçen popülerlik ölçümlerine göre en popüler programlama dilleri C/C++, Java ve C# dilleridir** ve Python’ın bu diller ile entegre bir şekilde genişletilebilmesini kolaylaştıran olanaklar mevcuttur. Dolayısıyla, bilimsel hesaplama veya programlamada bir adım öteye gidip bu dilleri de öğrenmek isteyen öğrenci, **adım adım bir geçiş** yapabilir.

⁶ Sage, <http://www.sagenb.org/> adresinde denenebilir. Ayrıca Dr. Kürsat Aker ve öğrencilerince Sage defter arayüzü ve temel yardım belgesi Türkçeleştirilmiştir. Türkçe arayüzlü yerel bir veya daha çok kamuya açık defter sunucusunun yakın bir gelecekte devreye alınması planlanmaktadır.

⁷ tutorial

Standart kütüphanenin genişliği Geliştiricileri Python için “pilleri dahil” sloganını kullanarak, standart kütüphanede her türlü iş için araçların bulunabileceğini vurgulamaktadırlar [11]. Bu, öğrencilere, hem akademik çalışmalarında hem de diğer işlerinde her türlü ihtiyaçları için tutarlı ve taşınabilir⁸ bir çerçeve sağlar.

Taşınabilir olması Derlenen bir dil olan Python'da yazılmış olan programlar, platformdan bağımsız olarak çalışabilirler (elbette bu, Python derleyicisinin ve ilgili kütüphanelerin o bilgisayarda bulunmasını gerektirir). Dolayısıyla herhangi bir şekilde çalıştırılabilir dosya oluşturmaya ve programı farklı platformlara uyarlamaya gerek kalmadan, programların öğrenciler ve hocalar arasında kolayca paylaşılabilmesi olanağı vardır.

Elbette, her konuda Python'ın avantajlı olduğunu öne sürmek mümkün değildir. Örneğin, bu kısımda Python'ın derlenen dillerden çok daha yavaş çalışması gibi noktalardan bahsedilmemiştir. Aslında belki de bu, lisans düzeyindeki bilimsel hesaplama açısından tek önemli dezavantaj olarak düşünülebilir. Öte yandan en kötü durumda çok vahim olan bu hız farkı, ortalama durumda ise derleme aşamasının olmamasıyla telafi edilebilen bir seviyededir. Python'ın bilimsel çalışmalarda kullanılması yönündeki bu argümanı tamamlayıcı nitelikteki tartışmalar için bkz. [12, 13, 14, 15].

2.2. Diğer Seçeneklerle Karşılaştırma

Yukarıda Python'ın bilimsel hesaplama eğitimi bağlamında taşıdığı olumlu özellikleri özetledik. Burada, Tablo 4 üzerinden özetleyerek, bilimsel hesaplama eğitiminde kullanılacak diğer seçenekler ile bir karşılaştırma yapıyoruz. Başlıca avantajların listelendiği bu karşılaştırmada üç alternatif ele alıyoruz: Düşük düzeyli ve derlenen diller (C/C++, Fortran, vs.), Python dışındaki yüksek düzeyli ve yorumlanan diller (Perl, Ruby, vs.) ve de özelleş-

miş bilimsel hesaplama ortamları (MATLAB, Mathematica, vs.).

3. Python Kullanan Proje ve Derslerden Örnekler

3.1. Bilimsel Projeler

Python, deneysel veri eldesi ve analizi veya benzetim gibi çeşitli amaçlı çalışmalarda küçük veya büyük ölçekli geliştirme projelerinde kullanılmıştır ve kullanılmaktadır. Büyük ölçekli projelerden iyi anlatılmış erken örneklerden birisi, Python'ın temellerini öğrenmek isteyenlere sıkça kaynak gösterilen *Python: Essential Reference* kitabının [16] yazarı Beazley'nin de katkıda bulunduğu olduğu Los Alamos Ulusal Laboratuvarları'nda geliştirilen moleküler dinamik sistemidir [17]. O çalışmada büyük bir yazılım projesi büyük oranda Python altyapısına kaydırılmış ve Python, tam da şimdiye kadar anlatılmaya çalışılan şekilde, benzetimden görselleştirmeye tüm işlerin ortak bir platformda ve kolayca yapılabilmesini sağlamıştır.

Görece daha küçük ve deneysel bir örnek olarak, ABD'nin Uzay Teleskobu Bilimi Enstitüsü'nün PyRAF çalışması alınabilir [18]. Bu projede, Fortran ve C gibi dillerde yazılan kodun Python ortamına entegrasyonunun örneği görülmektedir. Ayrıca çalışmalar esnasında çok büyük boyutlardaki verilerin sorunsuzca kullanılabilirliği, Python'ın çok geniş bir yelpazedeki projelere gerekli desteği sağladığına işaret etmektedir.

Python'ın, başka dillerle de desteklenerek, bilimsel çalışmalarda kullanımının çok sayıdaki örneği için *Computing in Science and Engineering* dergisinin özel Python sayısına veya *Python in Science* konferanslarının belgelerine danışılabilir.

3.2. Hesaplamalı Fizik Dersleri

Bu bölümde halihazırda Python'a geçiş yapmış çeşitli hesaplamalı fizik derslerine değinilerek bunlardaki tecrübelerin sunulması amaçlanmaktadır. Bu doğrultuda ele alınacak dersleri genel

hesaplamalı fizik ve özelleşmiş konular olarak iki kategoride ele almak mümkündür.

Genel dersler kategorisindeki en çarpıcı örnek, Prof. Landau'nun öncülüğünü yaptığı "Hesaplamalı Fizik" lisans programıdır [19]. Bu program, deney ve kuramın yanında hesaplamamanın üçüncü bilimsel yaklaşım olduğu fikri ve hesaplama becerisinin fizik gibi programların mezunlarının iş bulmalarını sağlayan başat etkenlerden olduğu gerçeğinden hareketle [20], dört yıllık eğitimi fizik, bilgisayar, matematik ve modelleme konuları üzerine oturtmuştur. Bu programın çeşitli derslerinde Python kullanılmaktadır; ki bunlar aynı zamanda faydalı kaynak teşkil ettikleri için ilgili bölümde daha geniş ele alınacaktır (bkz. Kısım 4.2). İngiltere ([21]) ve Almanya'da ([22]) da hesaplamalı bilim derslerinde Python'ın kullanıldığı örnekleri görmek mümkündür. Bu derslerde hem entegral almak gibi doğrudan sayısal yöntemler hem de hidrojen atomunun kuantum mekaniği gibi fiziksel problemler işlenmektedir; ki ikincisi için Python'ın görselleştirme yeteneklerinden çokça faydalanılmaktadır.

Özelleşmiş dersler arasında her ikisi de doğrusal olmayan ve çok serbestlik dereceli karmaşık sistemleri ele alan iki örnek beraberce verilebilir ([23, 24]). Bu iki derste de, geleneksel hesaplamalı bilim derslerinin dışında konular işlenmekte ve yine Python'ın görselleştirme becerilerinden faydalanılmaktadır.

4. Eğitim Kaynakları

Bu kısımda önce genel programlamacı/bilgisayar bilimci gözüyle Python'ın öğrenilebileceği kaynaklar; akabinde ise hesaplamalı bilimciye uyarlanmış Python kaynakları sıralanacaktır. Her iki grup kaynak için de İnternet'ten serbestçe ulaşılabilecek kaynaklar ayrı paragrafta ele alınacaktır.

4.1. Genel Python Kaynakları

Daha önce de belirtildiği gibi, Python en popüler programlama dillerinden birisidir. Keza,

birçok bilgisayar bilimleri bölümü, bilgisayar bilimine giriş derslerinde ('CS-1') kullanılacak dil için C veya Java gibi dillerden Python'a geçiş yapmanın yerinde bir seçim olduğunu öne sürmektedir [25]. Bu yaklaşım çerçevesinde yazılmış olan ve yaygın olarak kullanılan bir ders kitabı Zelle'nin kitabıdır [26]. Zelle, bu kitabın devamı niteliğindeki ve daha ileri konuları işlerken C++ dilini de öğretmeyi hedefleyen bir diğer kitabın da eşyazarıdır [27]. Ayrıca, programlamacılara Python dilini anlatmak amacıyla yazılmış ve ders kitabı niteliğinde olmayan başka birçok kaynak da vardır (örn., bkz. [28]).

Resmi dokümantasyon gibi İnternet'ten ücretsizce ulaşılabilen kitapların başlıcaları arasında, bilgisayar bilimine giriş niteliğindeki "*Think Python*"ı ([29]) ve başka bir dilde programlamayı bilenler için yazılmış olan "*Dive into Python*"ı ([30]) sayabiliriz. Her iki kaynak da, Python öğrenmek isteyenler için gayet nitelikli ve kapsamlı ders kitabı sağlamaktadırlar. Downey, "*Think Python*"da sıfırdan başlayarak programlamamanın temel bileşenlerini ele aldıktan sonra oldukça geniş bir şekilde nesne yönelimli programlamayı da işlemektedir. Bu kitabın dikkat çekici bir özelliği hata ayıklamaya yaptığı vurgudur. "*Dive into Python*" ise, amacına uygun bir şekilde, daha çok uygulamalar üzerinden Python'ı anlatmakta ve İnternet programlama ile ileri düzeydeki bazı konulara geniş yer ayırmaktadır.

4.2. Bilimsel Hesaplamada Python Kaynakları

Bilimsel veya mühendislik amaçlı programlama odaklı Python kitaplarının hepsi ders kitabı niteliğindedir ve ücretli olarak temin edilebilmektedirler. Bunların en önemli ve standart niteliğinde olanları, bilimsel hesaplamada Python kullanımının öncülerinden Hans Petter Langtangen tarafından yazılmış olanlarıdır. Langtangen, biri bilimsel hesaplamaya giriş amaçlı [31]; diğeri ise ileri düzeye yönelik [15] iki temel kaynak yazmıştır. Bunlardan ilki, Oslo Üniversitesi'ndeki tüm fen bilimleri bölümlerinin programlarında hesaplamayı

önemli bir bileşen haline getirmeyi hedefleyen reformun sonucu olarak ortaya çıkmıştır. Dolayısıyla bu ilk kitapta temel kavramların yanısıra değişik yöntemler ve yaklaşımlar ele alınmaktadır. İkincisinde ise, Python ile bilimsel hesaplama yapan bir bilimcinin ihtiyaç duyabileceği her türlü konu kapsamlı bir şekilde işlenmiştir. Ayrıca kitabın içeriği fiziksel bilimlerde çalışanlar için özellikle uygundur ve işlenen birçok konu ve örnek doğrudan çalışmalarda kullanılacak kodlara temel oluşturacak niteliktedir. Yukarıda da değinildiği gibi hız kazanma açısından Python'da bilimsel hesaplamaların önemli bir bileşeni olan Fortran ve C/C++ ile ikili programlama yöntemlerine de bu ikinci kitapta birer bölüm ayrılmıştır. Langtangen'in klasikleşmiş kitaplarının dışında, geleneksel bir mühendislikte sayısal yöntemler dersine yönelik ([32]) ve daha genel bilimsel betik programlamaya yönelik ([33]) iki kitap daha sayılabilir.

İnternet'ten ücretsiz olarak ulaşılabilecek kaynakların, Python'ın bilimsel hesaplama camiasının genişliği göz önüne alındığında, sayıca çok olmaları şaşırtıcı değildir. Öncelikle çok sayıda öğretici belgeye ve sunuma ulaşmak mümkündür (kapsamlı bir örnek için bkz. [34]). Bunun ötesinde, kitap mahiyetinde EuroSciPy ekibi tarafından hazırlanan "Python Scientific Lecture Notes" [35] ve örneğin "Python for Computational Science" [36] İnternet'ten indirilebilir. Alanındaki çeşitli temel soruları Python'ı kullanarak çözmeyi anlatan ve geleneksel sayısal fizik derslerinde işlenmeyen bazı konuyu içeren *Computational Modelling in Complexity Science* da İnternet'ten ücretsiz olarak indirilebilmektedir [24]. Bu kitap, [29]'ın devamı olacak şekilde tasarlanmıştır.

Bilimsel hesaplama odaklı eğitim kaynakları arasında diğer önemli kategori, İnternet üzerinden yayınlanan derslerdir. Bunlardan özellikle üçü Python öğrenenler için değerli kaynaklardır.

Software Carpentry *Software Carpentry* [37], geniş bir kapsama sahip olmakla beraber prog-

ramlama dili olarak Python'ı seçmiştir [37] ve bu sayede birçok farklı bağlamda Python'ın kullanımını işlemektedir. Bu konular, temel programlama, gelişkin veri yapıları, program testleri, nesne yönelimli programlama, düzenli ifadeler⁹, NumPy dizileri ile matris hesaplamaları, mul-timedya ve ağ programlamadır. *Software Carpentry* bilimcilere özgü bilgisayar eğitimi konusunda yıllardır geliştirilmekte olmanın avantajının yanında yılın çeşitli dönemlerinde kişilerin İnternet üzerinden kaydolup ücretsiz olarak takip edebileceği dönemlik bir program haline de getirilmiştir.

MIT OCW MIT'nin açık ders kaynakları arasında yayınlanan "Bilgisayar Bilimlerine ve Programlamaya Giriş" dersi [38], oturum videoları, ödevler, okumalar ve sınavlarla tam bir derse tekabül etmektedir. Dikkat çekici bir yanı, birçok dersin aksine kaynak olarak, yukarıda da sayılmış olan ücretsiz kitap ve belgeleri kullanıyor olmasıdır. Bu derste de, temel programlama kavramları ve nesne yönelimli programlamaların yanında çeşitli örneklerle beraber Python'da bilimsel hesaplama ele alınmaktadır. Verilen ödevler için destekleyici kaynak kodları da sitede bulunmaktadır.

A Survey of Computational Physics Yukarıda da adı geçmiş olan Prof. Landau'nun öncülüğünü yaptığı programın hesaplamalı fizik isimli ikinci dersinin İnternet sitesi [39], hem ilgili kitabın [40] Python'a uyarlanmış ve ücretsiz olarak indirilebilen sürümünü; hem de ders videolarını içermektedir. Bunun yanında aynı sitede ders yansılarını ve ödevleri de görmek mümkündür. Bu dersin iki çok önemli özelliği vardır. Birincisi, hesaplamalı fizik odaklı olmasından ötürü kaostan Feynman entegrallerine kadar birçok ilginç konuyu ve temel kavramlardan paralel hesaplamaya kadar birçok farklı yaklaşımı ele almasıdır. İkincisi ise, programlama konusu yapılan bu çok çeşitli problemlerin kaynak kodlarını Java, C, F95/77 ve Python olarak sunmasıdır.

⁹ regular expressions

5. Sonuç

Günümüzde, makine kaynaklarındansa insan kaynağının çok daha önemli olduğunu düşün-
müzde, Python, lisans seviyesindeki bilim-
sel hesaplama için en uygun dil gibi görünmek-
tedir. Burada “en uygun” ile kastedilen, elbette,
“her açıdan en iyi olmak” değil; tüm hususlar
ağırlandırılarak göz önünde bulundurulduğun-
da “optimum olmak”tır. Üniversitelerimizdeki
öğretim görevlileri, özgür yazılımın getirdiği
serbestlik ve zengin öğretim kaynaklarının sağ-
ladığı geniş seçeneklerden faydalanarak, ders-
lerinde kolayca Py- hon’a geçebilirler.

Kaynaklar

[1] Peter J. A. Cock and *et al.* Biopython: fre-
ely available Python tools for computational
molecular biology and bioinformatics. *Bio-
informatics*, 25(11):1422-1423, June 2009.

[2] <http://www.python.org/>.

[3] Guido van Rossum. Computer Program-
ming for Everybody (Revised Proposal):A
Scouting Expedition for the Programmers of
Tomorrow. Technical report, Corporation for
National Research Initiatives, 1999.

[4] A. M. Kuchling. Functional Programming
HOWTO — Python v2.6.5c2 documentation.
[http://www.python.org/doc/
current/howto/
functional.html](http://www.python.org/doc/current/howto/functional.html).

[5] <http://en.wikipedia.org/wiki/Simula>.

[6] D. P. Bischak and S. D. Roberts. Object-
oriented simulation. pages 194-203, 1991.

[7] [http://en.wikipedia.org/wiki/
Measuring_
programming_language_
popularity](http://en.wikipedia.org/wiki/Measuring_programming_language_popularity).

[8] [http://www.tiobe.com/index.php/
content/
paperinfo/tpci/index.html](http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html).

[9] <http://www.python.org/about/quotes/>.

[10] <http://www.sagemath.org/>.

[11] Paul F. Dubois. Guest Editor’s Introduc-
tion: Python: Batteries Included. *Computing in
Science & Engineering*, 9(3):7-9, 2007.

[12] Travis E. Oliphant. Python for Scienti-
fic Computing. *Computing in Science and En-
gineering*, 9(3):10-20, 2007.

[13] M. F. Sanner. Python: a programming
language for software integration and develop-
ment. *Journal of molecular graphics & mo-
delling*, 17(1):57-61, February 1999.

[14] Julius B. Lucks. Python - All a Scientist
Needs. Mar 2008.

[15] Hans P. Langtangen. *Python Scripting for
Computational Science*. Springer, 3rd ed. edi-
tion, 2010.

[16] David M. Beazley. *Python Essential Refe-
rence (4th Edition)*. Addison-Wesley Profes-
sional, 2009.

[17] David M. Beazley and Peter S. Lomdahl.
Feeding a Large-scale Physics Application to
Python. In *In 6th International Python Con-
ference*, pages 21-28, 1997.

[18] Perry Greenfield. Reaching for the Stars
with Python. *Computing in Science and En-
gineering*, 9(3):38-40, 2007.

[19] [http://www.physics.oregonstate.edu/~ru-
bin/CPUG/index.html](http://www.physics.oregonstate.edu/~rubin/CPUG/index.html).

[20] Osman Yasar and Rubin H. Landau. Ele-
ments of Computational Science and Enginee-
ring Education. *SIAM Review*, 45(4):787-805,
2003.

[21] P. Borchers. Python: a language for com-
putational physics. *Computer Physics Com-
munications*, 177(1-2):199-201, July 2007.

- [22] Arnd Backer. Computational Physics Education with Python. *Computing in Science & Engineering*, 9(3):30-33, 2007.
- [23] Christopher R. Myers and James P. Sethna. Python for Education: Computational Methods for Nonlinear Systems. *Computing in Science and Engineering*, 9(3):75-79, 2007.
- [24] Allen B. Downey. Computational Modeling and Complexity Science. <http://greenteapress.com/compmo/>.
- [25] Bradley Miller, D. Ph, and David Rannum. *Python Programming in Context*. Jones & Bartlett Publishers, 2008.
- [26] John Zelle. Python Programming: An Introduction to Computer Science 2nd Ed. Franklin, Beedle & Associates Inc., 2010.
- [27] David M. Reed and John Zelle. *Data Structures and Algorithms: Using Python and C++*. Franklin Beedle & Associates.
- [28] Mark Lutz. Learning Python: Powerful Object-Oriented Programming. *O'Reilly Media*, 2009.
- [29] Allen B. Downey. Python for Software Design: How to Think Like a Computer Scientist. *Cambridge University Press*, 2009.
- [30] Mark Pilgrim. *Dive Into Python*. Apress, 2004.
- [31] Hans P. Langtangen. A Primer on Scientific Programming with Python (Texts in Computational Science and Engineering). *Springer*, 1 edition, September 2009.
- [32] Jaan Kiusalaas. *Numerical Methods in Engineering with Python*. Cambridge University Press, 2 edition, January 2010.
- [33] Drew McCormack. *Scientific Scripting with Python*. lulu.com, November 2009.
- [34] Python for High Performance and Scientific Computing. <http://www.mcs.anl.gov/~wscullin/python/tut/sc09/Site/Introduction.html>.
- [35] <http://code.google.com/p/esdanalysisistools/downloads/list>.
- [36] Hans Fangohr, Jacek Generowicz, and Thomas Fischbacher. Python for Computational Science. www.soton.ac.uk/~fangohr/computing/Python4ScientificComputation.pdf, 2009.
- [37] <http://software-carpentry.org/>.
- [38] <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00-introduction-to-computer-science-and-programming-fall-2008/>.
- [39] A Survey of Computational Physics. <http://www.physics.oregonstate.edu/~rubin/Books/Survey/index.html>.
- [40] Rubin H. Landau, Jose Paez, and Cristian C. Bordeianu. *A Survey of Computational Physics: Introductory Computational Science*. Princeton University Press, 2008.

Kategori	Kütüphane	Kısa Açıklama
Temel Kütüphaneler	Numpy Scipy	Hızlı dizi veriyapısı ve ilgili işlemler Temel bilimsel işlevler (sayısal entegrasyon, dönüşümler, istatistik, vs.)
	Matplotlib	Kapsamlı 2B görselleştirme; 3B görselleştirme

Tablo 1: Bilimsel hesaplama için temel Python kütüphaneleri

Kategori	Kütüphane	Kısa Açıklama
Çeşitli bilimsel kütüphaneler	BioPython Brian SymPy SfePy	Biyo-enformatik Yapay sinir ağları Sembolik hesaplama Sonlu elemanlar
Başka dil çağırma	f2py Cython RPy2	Fortran kodu çağırma için C kodu çağırma için R kodu çağırma için
Başka dildeki bilimsel kütüphanelere bağlantı	pygs1 PyROOT	GNU bilimsel kütüphanesi için CERN'in ROOT çerçevesi için
Paralel hesaplama	mpi4py PyCUDA PyOpenCL	MPI işlevleri için CUDA ile GPU kodlama için OpenCL ile GPU kodlama için

Tablo 2: Bilimsel hesaplama için temel üç kütüphanenin dışındaki Python kütüphaneleri/araçları

Kategori	Geliştirme Ortamı
Genel geliştirme ortamları	IDLE Eclipse Eric Open Komodo
MATLAB tarzı ortamlar	Spyder Python Toolkit IEP
'Defter' tarzı ortamlar	Sage Ipython (sürüm > 0.12) Notebook

Tablo 3: Bilimsel hesaplama için platformlar arası ve özgür Python geliştirme ortamları

Derlenen Diller (C, Fortran, vb.)	Yorumlanan Diller (Perl, Ruby, vb.)	Hesaplama Ortamları (Matlab, Mathematica, vb.)
Öğrenmesi kolay Etkileşimli Yüksek seviyeli Geniş standart kütüphane Taşınabilirlik Sage	Kapsamlı bilimsel kütüphaneler Akademide yaygın kullanım Genel olarak yaygın kullanım Geniş standart kütüphane Geniş eğitim kaynakları Sage	Özgür, lisans sorunu yok Genel amaçlı Kaynak kodu görebilme Taşınabilirlik

Tablo 4: Bilimsel hesaplama eğitimi için Python'ın diğer seçeneklere kıyasla avantajları