FİZ433 FİZİKTE BİLGİSAYAR UYGULAMALARI

(DERS NOTLARI)

Hazırlayan:

Prof.Dr. Orhan ÇAKIR

Ankara Üniversitesi, Fen Fakültesi, Fizik Bölümü

Ankara, 2017

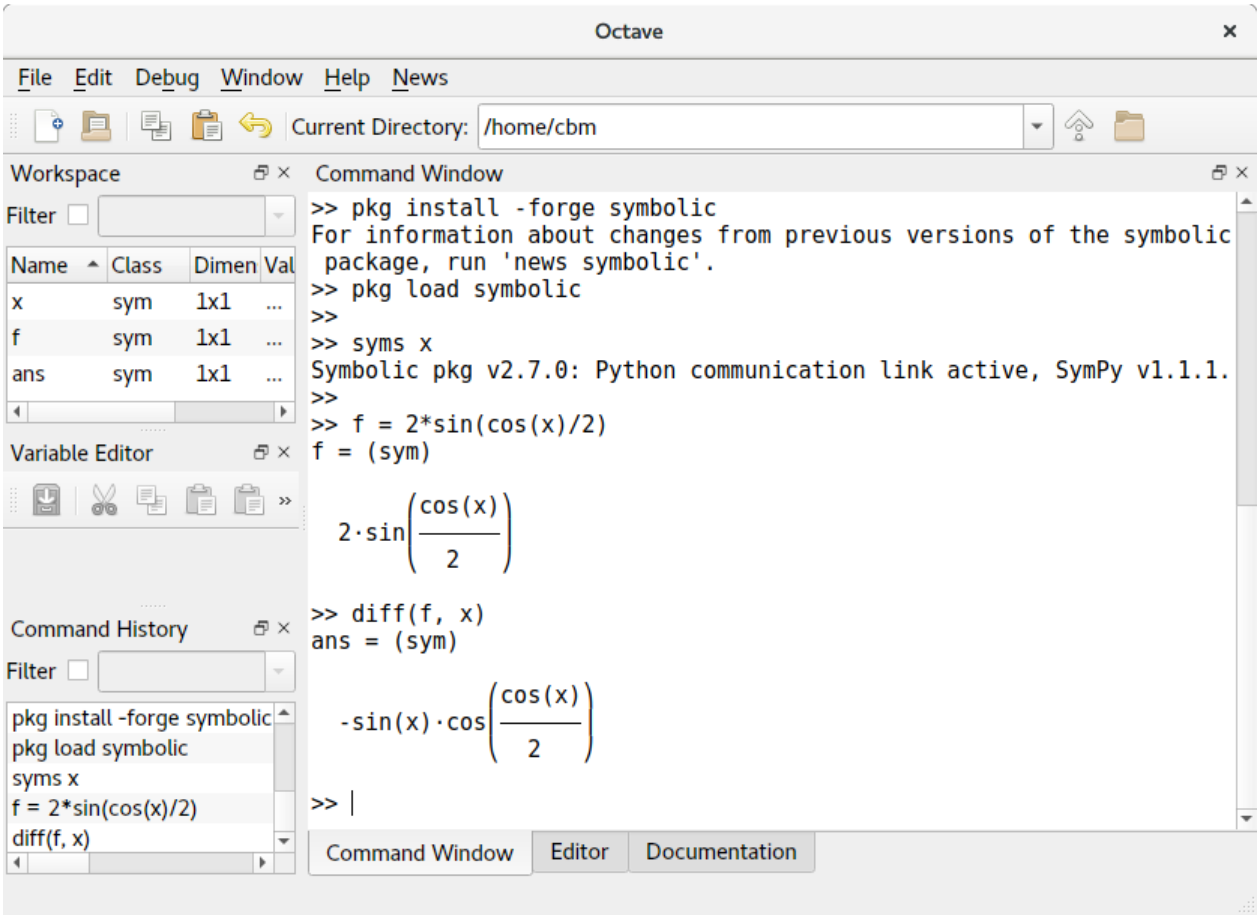
İÇİNDEKİLER

KONU 14

FİZİKTE SEMBOLİK HESAPLAMA II

Sembolik hesaplama, matematiksel nesnelerin sembolik olarak hesaplanması ile ilgilidir. Bu, matematiksel nesnelerin tam olarak temsil edilmediği, yaklaşık olarak ifade edilmediği ve değerlenmemiş değişkenlere sahip matematiksel ifadelerin sembolik biçimde bırakıldığı anlamına gelir.

Octave da sembolik hesap komut penceresinden yapılabilir. Sembolik paket kurulumu yaptıktan sonra çalışmak için load komutu ile yükleme yapılır.



Şekil 14.1 Octave komut penceresi ve sembolik hesap

Octave sembolik hesap için SymPy paketini kullanır.

* Serbest: BSD altında lisanslıdır, SymPy bir serbest yazılımdır.

Python-temelli: SymPy tamamen Python da yazılmıştır, ve dil olarak Python kullanır.

Hafif: SymPy sadece mpmath e bağlıdır, bu bir kayan nokta aritmetik Python kütüphanesidir, kullanımı kolaydır.

Bir kütüphane: etkileşimli kullanımı dışında, SymPy diğer uygulamalar içine de eklenebilir, ve istenen fonksiyonlarla genişletilebilir. Örneğin, açık kaynak kodlu SageMath matematik sistemi SymPy içerir.

SymPy güçlü bir sembolik hesaplama sistemi (bu arada sıklıkla bilgisayar cebir sistemi veya sadece CAS'lar olarak da adlandırılır), değişkenlerle sembolik ifadeleri hesaplayabilir.

Daha sonra göreceğimiz gibi, SymPy'de değişkenler semboller kullanılarak tanımlanır. Birçok sembolik manipülasyon sisteminin aksine, SymPy'deki değişkenler kullanılmadan önce tanımlanmalıdır.

Burada x + 2y matematiksel ifadesini temsil eden sembolik bir ifade tanımlayalım. Bilgisayarda terminalde 'python' komutu ile Python çalıştırılır. Aşağıdaki ifadeler gelen komut satırında yazılır:

```
>>> from sympy import symbols
>>> x, y = symbols('x y')
>>> expr = x + 2*y
>>> expr
x + 2*y
```

Burada x ve y sembolik değişkenler. Aşağıdaki gibi bir hesap yaparsak:

```
>>> expr + 1
x + 2*y + 1
>>> expr - x
2*y
```

elde ederiz. Burada 'expr - x' ifadesinde cevap olarak 'x + 2*y +1 - x' ifadesi değil '2*y' vermiştir. Benzer şekilde 'sqrt(8)' sayısal ifadesi de '2*sqrt(2)' olarak alınmaktadır.

SymPy de sembolik ifadeler bir formdan başka bir forma dönüştürülebilir, örneğin:

```
>>> from sympy import expand, factor
>>> expanded_expr = expand(x*expr)
>>> expanded_expr
x**2 + 2*x*y
>>> factor(expanded_expr)
x*(x + 2*y)
```

SymPy ifadeleri basitleştirir, türevleri hesaplar, integraller ve limit hesabı yapabilir, denklemleri çözebilir, matrislerle çalışabilir ve çok daha fazlasını yapabilir ve hepsini sembolik olarak yapabilir. Aşağıda bazı örnekler verilmiştir:

Sembolik değişken tanımlama:

```
>>> from sympy import *
>>> x, t, z, nu = symbols('x t z nu')
```

Unicode karakterlerle yazma:

```
>>> init_printing(use_unicode=True)
```

İfadenin türevini alma:

```
>>> diff(sin(x)*exp(x), x)
 x           x
e ·sin(x) + e ·cos(x)
```

İfadenin integralini alma:

```
>>> integrate(exp(x)*sin(x) + exp(x)*cos(x), x)
```

```
 x
e ·sin(x)
```

Sınırlı integral hesaplama:

```
>>> integrate(sin(x**2), (x, -oo, oo))
√2·√π
────
  2
```

Limit alma:

```
>>> limit(sin(x)/x, x, 0)
1
```

Basit denklem çözme:

```
>>> solve(x**2 - 2, x)
[-√2, √2]
```

Diferensiyel denklem çözme:

```
>>> y = Function('y')
>>> dsolve(Eq(y(t).diff(t, t) - y(t), exp(t)),
y(t))
            -t    ⎛     t⎞   t
y(t) = C₂·e   + ⎜C₁ + ─⎟·e
                ⎝     2⎠
```

Matrisin özdeğerlerini bulma:

```
>>> Matrix([[1, 2], [2, 2]]).eigenvals()
⎧3   √17      3   √17    ⎫
⎨─ - ────: 1, ─ + ────: 1⎬
⎩2    2       2    2     ⎭
```

Bessel fonksiyonlarını küresel Bessel fonksiyonları cinsinden yazma:

```
>>> besselj(nu, z).rewrite(jn)
√2⋅√z⋅jn(ν - 1/2, z)
────────────────────
        √π
```

Değişkenlere değer atama:

```
>>> expr = x**3 + 4*x*y - z
>>> expr.subs([(x, 2), (y, 4), (z, 0)])
40
```

Integral ifadesini LaTeX formatında yazma:

```
>>> latex(Integral(cos(x)**2, (x, 0, pi)))
\int\limits_{0}^{\pi} \cos^{2}{\left(x \right)}
\, dx
```

$$\int_0^{\pi} \cos^2 (x)\, dx$$

Bu alanda birçok bilgisayar cebir sistemi var. SymPy'yi alternatiflerden daha iyi bir seçim yapan nedenler bulunmaktadır.

Öncelikle, SymPy tamamen ücretsizdir. Açık kaynak kodludur ve BSD lisansı altında lisanslıdır, böylece kaynak kodu değiştirilebilir.

İkincisi, SymPy paketi Python'u kullanır. Çoğu bilgisayar cebir sistemi kendi dilini icat eder. SymPy böyle değildir, tamamen Python ile yazılmıştır. Bu, zaten Python'u biliyorsanız, SymPy ile çalışmaya başlamanın çok daha kolay olduğu anlamına gelir, çünkü sözdizimini zaten biliyorsunuzdur (ve Python'u bilmiyorsanız, öğrenmesi gerçekten kolaydır). Python'un iyi tasarlanmış, zorlu testi yapılmış bir dil olduğunu zaten biliyoruz. SymPy geliştiricileri, matematiksel yazılım yazma becerilerinden emindir, ancak programlama dil tasarımı tamamen farklı bir şeydir. Varolan bir dili yeniden kullanarak, önemli olan şeylere odaklanabiliriz: matematik.

Başka bir bilgisayar cebir sistemi olan Sage, Python'u da dili olarak kullanır. Ancak Sage, büyük bir gigabayt indirme gerektirir. SymPy'nin bir avantajı hafif olmasıdır. Nispeten küçük olmasının yanı sıra, Python'dan başka bir bağımlılığı yoktur, bu yüzden hemen hemen her yerde kolayca kullanılabilir. Ayrıca, Sage'in hedefleri ve SymPy'nin amaçları farklıdır. Sage, matematik için tam özellikli bir sistem olmayı ve tüm büyük açık kaynaklı matematik sistemlerini bir araya getirerek bunu gerçekleştirmeyi hedeflemektedir. Sage'de bütünleştirme gibi bir işlev çağırdığınızda, içerdiği açık kaynak paketlerinden birini çağırır. Aslında, SymPy Sage'e dahil edilmiştir. SymPy ise SymPy'de uygulanan tüm özelliklerle bağımsız bir sistem olmayı hedeflemektedir.

SymPy'nin son önemli özelliği, bir kütüphane olarak kullanılabiliyor olmasıdır. Çoğu bilgisayar cebir sistemi etkileşimli ortamlarda kullanılabilir olmaya odaklanır, ancak bunları otomatikleştirmek veya genişletmek istiyorsanız bunu yapmak biraz çaba ister. SymPy ile etkileşimli bir Python ortamında kolayca kullanabilirsiniz veya kendi Python uygulamanıza içe aktarabilirsiniz. SymPy ayrıca kendi özel işlevlerinizle genişletmeyi kolaylaştırmak için API'ler de sunar.

Kaynaklar:

[1] https://www.sympy.org/en/index.html, Erişim tarihi: 08.05.2017.

[2] https://live.sympy.org, Erişim tarihi: 08.05.2017.