مثال آموزشی : مدلسازی مبتنی بر معادلات از خواص گاز کامل

بخش اول: مدل گاز کامل

یکی از روشهای محاسبه خواص واقعی گازها، تعریف یک مسیر فرضی مبتنی بر کاهش فشار گاز تا حد گاز ایدآل میباشد (جهت مطالعه رجوع شود به مرجع [1]، بخش ۵، شبکه ترمودینامیک)، با این روش میتوان خواص گاز را به خواص همان گاز در شرایطی که معادله حالت آن با گاز ایدآل مطابقت داشته باشد نسبت داد و از این ترفند به منظور محاسبه خواص یک گاز واقعی استفاده کرد . بعلاوه بسیاری از ضرایب مهم و مورد نیاز در محاسبات خواص گازها تنها در شرایط گاز ایدآل در دسترس میباشند و نرم افزار باید توانایی استفاده از این ضرایب برای محاسبه مقادیر درست در شرایط واقعی را داشته باشد . توانایی در محاسبه دقیق خواص گاز ایدآل لازمه هر نرم افزار شبیه سازی فرآیندی میباشد .

ماژول ذیل مدلی مبتنی بر معادلات برای رفتار یک گاز کامل تعریف میکند. این مدل دارای ۱۱ معادله میباشد که در دو حالت تقریب چندجمله ای و هایپربولیک از تغییرات ظرفیت گرمایی، خواص گاز را تعیین میکنند . این ۱۱ معادله شامل ۸ معادله مستقل میباشند . تعداد متغییرهای اصلی و کمکی جمعا ۱۰ عدد میباشد که دستگاه معادلات را میتوان با تعیین ۲ متغییر برای محاسبه ۶ مجهول حل نمود.

```
module IdealGasEos
  #Units J,Kmol,Kelvin
  #انواع داده های داخلی دانا را استفاده میکنیم
  using DanaTypes
  #مدل صادره شامل یک ساختمان داده و رویه های مربوطه میباشد
  export DANAIdealGasEos,setEquationFlow
  #ساختمان داده مدل گاز ایدآل
  type  DANAIdealGasEos <: DanaModel
  #مولد ساختمان
      DANAIdealGasEos()=begin
        #مقادیر عناصر مدل به ترتیب تعریف شدن
        new(8314.4621,"",true,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,
          #معادلات مدل
          #معادلات مدل جمعا ۱۱ معادله میباشد که ۸ معادله مستقلند
          [
            :(P*v=R*T),#معادله حالت
            :(Cp=C1+C2*T+C3*T^2+C4*T^3+C5*T^4),#Poly Cp
```

```
            :(Cp=C1+C2*((C3/T)/sinh(C3/T))^2+C4*((C5/T)/cosh(C5/T))^2),#Hyper Cp

            :(ICpOnTDT=C2*T+(C3*T^2)/2+(C4*T^3)/3+(C5*T^4)/4+C1*log(T)),#Integral of Cp/T
Poly

            :(ICpOnTDT=(C2*C3*coth(C3/T)+C1*T*log(T)+C4*T*log(cosh(C5/T))-
C2*T*log(Sinh(C3/T))-C4*C5*tanh(C5/T))/T),#Integral of Cp/T Hyper

            :(Cv=Cp-R),#Cv Def

            :(ICpDT=C1*T+1/60*T^2*(30*C2+T*(20*C3+3*T*(5*C4+4*C5*T)))),#Integ of Cp Poly

            :(ICpDT=C1*T+C2*C3*coth(C3/T)-C4*C5*tanh(C5/T)),#Integ of Cp Hyper

            :(u=ICpDT-R*T), #Internal energy dep

            :(h=ICpDT), #Enthalpy def

            :(s=ICpOnTDT-R*log(P)) #Entropy

          ],Array(Expr,0)

        )

      end

      # پارامتر های مدل قبل از حل مشخص میباشند

      R::Float64 #پارامتر ثابت R

      CASNO::String #کد اختصاری ماده

      usePolynomialEstimationOfCp::Bool # کدام تقریب برای محاسبه ظرفیت گرمایی استفاده
شود

      C1::Float64 #ضریب

      C2::Float64 #ضریب

      C3::Float64 #ضریب

      C4::Float64 #ضریب

      C5::Float64 #ضریب

      # متغییر های مدل

      #۸ تعداد

      v::Float64 #حجم مخصوص

      T::Float64 #دما

      P::Float64 #فشار

      Cp::Float64 #ظرفیت گرمایی

      Cv::Float64 #ظرفیت گرمایی حجم ثابت
```

```
      u::Float64 #انرژی داخلی

      h::Float64 #آنتالپی

      s::Float64 #آنتروپی

      # متغییرهای وابسته

      #۲ تعداد

      ICpOnTDT::Float64 #Integral of Cp/T Poly

      ICpDT::Float64 #Integral of Cp/T Poly

      #تعداد کل متغییرها ۱۰ میباشد

      equations::Array{Expr,1} #آرایه ای برای نگه داری کلیه معادلات لازم

      equationsFlow::Array{Expr,1} #آرایه ای از معادلات حاکم

  end

  #با توجه به شرایط مدل ۸ معادله مستقل از ۱۱ ملادله برگزیده میشود

  function setEquationFlow(this::DANAIdealGasEos)

    if this.usePolynomialEstimationOfCp

this.equationsFlow=[this.equations[1],this.equations[2],this.equations[4],this.equations[6]
,this.equations[7],this.equations[9],this.equations[10],this.equations[11]]

    else

this.equationsFlow=[this.equations[1],this.equations[3],this.equations[5],this.equations[6]
,this.equations[8],this.equations[9],this.equations[10],this.equations[11]]

    end

  end

end
```

بخش دوم: محاسبه ظرفیت گرمایی

ظرفیت گرمایی یک گاز کامل نقش بسیار مهمی در محاسبه خواص ترمودینامیکی گازها (آنتروپی، آنتالپی، انرژی داخلی و ...) ایفا میکند و لذا محاسبه دقیق آن برای رسیدن به دقت بالا در محاسبات لازم است .هندبوک پرری [2] یکی از مراجع اصلی برای استخراج خواص ترمودینامکی مواد به حساب می آید .ماژول فوق ضرایب مربوط به دو تقریب چندجمله ای و هایپربولیک را از آخرین نسخه هندبوک پرری (ویرایش هشتم) برای یک ماده خاص برمیگرداند .

```
# استخراج ضرایب مناسب براي دو روش محاسبه ظرفیت گرمایي با استفاده از تقریب چند جمله اي
یا هایپربولیک
```

```
module CpIdeal

# توابع صادره از ماژول

  export C0Poly,C0Hyper

#فايل هاي جداول باز شوند


cp_polynomial=open("./share/julia/site/v0.2/CpIdeal.jl/src/Tables/perryHeatCapIdealGas(Table2-155).table");


cp_hyperbolic=open("./share/julia/site/v0.2/CpIdeal.jl/src/Tables/perryHeatCapIdealGas(Table2-156).table");

#جدول ۲-۱۵۵ هندبوک پري در ماتريس بارگذاري شود

  data_poly,header_poly=readdlm(cp_polynomial,';',has_header=true);

#جدول ۲-۱۵۶ هندبوک پري در ماتريس بارگذاري شود

  data_hyper,header_hyper=readdlm(cp_hyperbolic,';',has_header=true);

#نياز به فايل نيست

  close(cp_polynomial)

  close(cp_hyperbolic)

#ثابت هاي تقريف چند جمله اي را براي يک ماده با کد مشخص برميگرداند

  function C0Poly(CasNo::String)

    length=size(data_poly)[1]

    i=1

    while (i<=length && data_poly[i,4]!=CasNo)

      i+=1;

    end

    if (i<=length)

      return (data_poly[i,6],data_poly[i,7],data_poly[i,8],data_poly[i,13],data_poly[i,14])

    end

  end

#ثابتهاي تقريب هايپربوليک را براي يک ماده با کد مشخص برميگرداند

  function C0Hyper(CasNo::String)

    length=size(data_hyper)[1]
```

```
    i=1
    while (i<=length && data_hyper[i,4]!=CasNo)
      i+=1;
    end
    if (i<=length)
      return
(data_hyper[i,6],data_hyper[i,7],data_hyper[i,8],data_hyper[i,9],data_hyper[i,10])
    end
  end
end
```

بخش سوم : تحلیل مدل

مثال زیر نحوه مدلسازی و تحلیل را در شبیه ساز دانا نمایش میدهد نرم افزار این توانایی را دارد که از مدلهای تعریف شده یک متغییر مدل بسازد کاربر به تعدادی از متغییرها مقدار تخصیص میدهد، نرم افزار متغییرها و پارامترهای معلوم را در معادلات جایگذاری میکند، معادلات ساده سازی میشوند. سپس دستگاه معادلات خطی تشکیل میشود، دستگاه معادلات تحلیل میگردد. هرگاه تعدادی از مجهولات محاسبه گردند اما مدل به طورکامل معلوم نشده باشد، معادلات به روز رسانی میشوند و این فرآیند تکرار میگردد .

```
module test
  #ماژول هاي مورد استفاده
  using HelperEquation #توابع لازم براي تحليل مدل
  using IdealGasEos #مدل گاز كامل
  using CpIdeal #ظرفيت حرارتي گاز كامل
  using Calculus #ماژول شامل توابع كلي در جبر
  function testforIdealGasModelWithCp()
    #يک مدل جديد از گاز كامل ساخته شود
    DNIdel=DANAIdealGasEos()
    #دو مقدار از متغيرها مشخص گردد
    DNIdel.P=12.0
    DNIdel.T=120.0
    #پارامتر نوع ماده
```

```
    DNIdel.CASNO="95-63-6" #Trimethylbenzene

    #پارامتر نوع تقريب ظرفيت گرمايي

    DNIdel.usePolynomialEstimationOfCp=true

    #از ماژول ظرفيت حرارتي ضرايب براي اين گاز استخراج ميشود

    DNIdel.C1,DNIdel.C2,DNIdel.C3,DNIdel.C4,DNIdel.C5 = C0Poly("95-63-6")

    #از ماژول مدل گاز كامل معادلات حاكم تعيين مي شود

    setEquationFlow(DNIdel)

    #از ماژول تحليل مدل متغيرها و پارامترهاي معلوم جايگذاري ميشوند

    a=replace(DNIdel)

    #معادله ۱ چاپ شود

    println(a[1]) #=> :(-(*(12.0,v),*(8314.4621,120.0)))

    # كليه معادلات ساده شود

    a=map(simplify,a)

    #معادله ۱ چاپ شود

    println(a[1]) #=> :(-(*(12.0,v),997735.452))

    #يک دستگاه معادلات جبري تشكيل شود

    vals,vars=analysis(a)

    println(a) #=>

    #:(-(*(12.0,v),997735.452))

    #:(-(Cp,78910.79999999999))

    #:(-(ICpOnTDT,211746.4556136655))

    #:(-(Cv,-(Cp,8314.4621)))

    #:(-(ICpDT,6.785928e6))

    #:(-(u,-(ICpDT,997735.452)))

    #:(-(h,ICpDT))

    #:(-(s,-(ICpOnTDT,20660.662161700304)))

    println(vals) #=>

#12     0       0       0       0       0       0       0       -997735.452

#0      1       0       0       0       0       0       0       -78910.79999999999

#0      0       1       0       0       0       0       0       -211746.4556136655
```

```
#0       -1      0       1       0       0       0       0       8314.4621

#0       0       0       0       1       0       0       0       -6785928

#0       0       0       0       -1      1       0       0       997735.452

#0       0       0       0       -1      0       1       0       0

#0       0       -1      0       0       0       0       1       20660.662161700304

    println(vars) #=>

#v

#Cp

#ICpOnTDT

#Cv

#ICpDT

#u

#h

#s

#constant

    #محاسبه ماتريس سطري پلكاني كاهش يافته -گوس جردن

    rVls=rref(vals)

    println(rVls) #=>

# 1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  -83144.6

# 0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  -78910.8

# 0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  -211746.0

# 0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  -70596.3

# 0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  -6.78593e6

# 0.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  -5.78819e6

# 0.0  0.0  0.0  0.0  0.0  0.0  1.0  0.0  -6.78593e6

# 0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  -191086.0

    #ستون آخر ماتريس سطري پلكاني مقدار مجهولات است

    DNIdel.v=-1*last(rVls[1,:])

    DNIdel.Cp=-1*last(rVls[2,:])

  end
```

```
  #توابع سطح بالاتر به منظور حل و به روز رساني تعريف ميشوند
  function testUpdateAndSolve()
    DNIdel=DANAIdealGasEos()
    DNIdel.P=12.0
    DNIdel.T=120.0
    DNIdel.CASNO="95-63-6"
    DNIdel.usePolynomialEstimationOfCp=true
    DNIdel.C1,DNIdel.C2,DNIdel.C3,DNIdel.C4,DNIdel.C5 = C0Poly("95-63-6")
    setEquationFlow(DNIdel)
    #جايگذاري، ساده سازي و تحليل در يک تابع خلاصه شده اند
    rVls,vars=solve(DNIdel)
    #و نتايج بطور در مدل به روز مي شوند
    update!(DNIdel,rVls,vars)
    #ميتوان بررسي کرد آيا اين نتايج در معادلات صادقند
    a=replace(DNIdel)
    println(map(eval,a)) #=>
#0
#0
#0
#0
#0
#0
#0
#0
  end
  #اگر دما مجهول باشد چه اتفاق خواهد افتاد
  function testIDealGas()
    ######Temprature is undef#######
    DNIdel=DANAIdealGasEos()
    DNIdel.P=2000.0
```

```
    DNIdel.v=4000.0

    DNIdel.CASNO="95-63-6"

    DNIdel.usePolynomialEstimationOfCp=true

    DNIdel.C1,DNIdel.C2,DNIdel.C3,DNIdel.C4,DNIdel.C5 = C0Poly("95-63-6")

    setEquationFlow(DNIdel)

    somthingUpdated=true

    fullDetermined=false

    #با توجه به اینکه دما مجهول است در مرحله اول ۳ معادله غیر خطي مي باشند

    #در مرحله اول تنها دما محاسبه خواهد شد

    #پس از محاسبه دما و به روز رساني مدل ميتوان كليه متغيرها را به دست آورد

    #در مرحله دوم تمام متغيرها محاسبه ميشوند

    while (somthingUpdated && !fullDetermined)

      rVls,vars=solve(DNIdel)

      somthingUpdated,fullDetermined=update!(DNIdel,rVls,vars)

    end

    dump(DNIdel) #=>

#DANAIdealGasEos

#  R: Float64 8314.4621

#  CASNO: ASCIIString "95-63-6"

#  usePolynomialEstimationOfCp: Bool true

#  C1: Float64 35652.0

#  C2: Float64 323.89

#  C3: Float64 0.305

#  C4: Float64 0.0

#  C5: Float64 0.0

#  v: Float64 4000.0

#  T: Float64 962.1789003043262

#  P: Float64 2000.0

#  Cp: Float64 629657.5360577751

#  Cv: Float64 621343.0739577751
```

```
#  u: Float64 2.6679239194320917e8

#  h: Float64 2.747923919432092e8

#  s: Float64 634526.1472890818

#  ICpOnTDT: Float64 697723.5627147412

#  ICpDT: Float64 2.747923919432092e8

#........

  end

  #درصورتي كه ترم غير خطي از معادلات خطي محاسبه نگردد

  function testIDealGasForNonlinearSolver()

    ######Temprature is undef#######

    DNIdel=DANAIdealGasEos()

    DNIdel.P=2000.0

    DNIdel.Cp=629657.0

    DNIdel.CASNO="95-63-6"

    DNIdel.usePolynomialEstimationOfCp=true

    DNIdel.C1,DNIdel.C2,DNIdel.C3,DNIdel.C4,DNIdel.C5 = C0Poly("95-63-6")

    setEquationFlow(DNIdel)

    somthingUpdated=true

    fullDetermined=false

    #در مرحله اول تنها ظرفيت گرمايي حجم ثابت محاسبه خواهد شد

    #در مرحله دوم اتفاق بيشتري نميافتد

    #تحليل بيشتر به توسعه تحليلگر به منظور حل معادلات غير خطي نياز دارد

    while (somthingUpdated && !fullDetermined)

      rVls,vars=solve(DNIdel)

      println("************one solution done************")

      somthingUpdated,fullDetermined=update!(DNIdel,rVls,vars)

    end

    dump(DNIdel)

  end

end
```

مراجع :

[1] : Engineering and Chemical Thermodynamics , Milo D. Koretsky , Oregon State University , 2nd Edition

[2] : PERRY'S CHEMICAL ENGINEERS' HANDBOOK EIGHTH EDITION