This is comment list on Form Factors & Structure factors available in the Irena package

Content

Form Factor	Parameter(s)
spheroid	**AspectRatio = ParticlePar1**
Integrated_Spheroid	**AspectRatio=ParticlePar1**
Cylinders	**Length=ParticlePar1**
CylindersAR	**AspectRatio=ParticlePar1**
Unified_Disc	**thickness = ParticlePar1**
Unified_Rod	**length = ParticlePar1**
Unified_RodAR	**AspectRatio = ParticlePar1**
Unified_Sphere	**none needed**
Algebraic_Globules	**AspectRatio = ParticlePar1**
Algebraic_Rods	**AspectRatio = ParticlePar1, AR > 10**
Algebraic_Disks	**AspectRatio = ParticlePar1, AR < 0.1**
User	**uses user provided functions.** **There are two user provided functions necessary -** **F(q,R,par1,par2,par3,par4,par5)** **and V(R,par1,par2,par3,par4,par5)** **the names for these need to be provided in strings...** **the input is q and R in angstroms**
Tube	**length=ParticlePar1** **//length in A** **WallThickness=ParticlePar2 //in A** **CoreRho =ParticlePar3** **//rho [10^10 cm-2] (not delta rho squared!!!) of core**

	material **ShellRho =ParticlePar4** **//rho [10^10 cm-2] (not delta rho squared!!!) of shell material** **SolventRho =ParticlePar5** **//rho [10^10 cm-2] (not delta rho squared!!!) of surrounding medium (air=0)**
CoreShell	**CoreShellThickness=ParticlePar1** **//skin thickness in Angstroms** **CoreRho =ParticlePar2** **//rho [10^10 cm-2] (not delta rho squared!!!) of core material** **ShellRho =ParticlePar3** **//rho [10^10 cm-2] (not delta rho squared!!!) of shell material** **SolventRho =ParticlePar4** **//rho [10^10 cm-2] (not delta rho squared!!!) of surrounding medium (air=0)**
Fractal aggregate	**FractalRadiusOfPriPart=ParticlePar1** **//radius of primary particle** **FractalDimension=ParticlePar2** **//Fractal dimension**

Structure factors included

Interferences	**reference:** Beaucage, G. (1995). J Appl Crystallogr **28**, 717-728. **Par1: ETA (center-to-center distance** **Par2: Pack (number of particles In nearest neighbor sphere)**
Hard Spheres	**reference: Percus-Yevick model, PERCUS,YEVICK PHYS. REV. 110 1 (1958), THIELE J. CHEM PHYS. 39 474 (1968), WERTHEIM PHYS. REV. LETT. 47 1462 (1981)** **Par1: Radius [A]** **Par2: Volume fraction (fraction)**
Square Well	**reference: SHARMA,SHARMA, PHYSICA 89A,(1977),212, NOTE - depths >1.5kT and volume fractions > 0.08 give UNPHYSICAL RESULTS when compared to Monte Carlo simulations** **Par1: Radius [A]** **Par2: Volume fraction [fraction]** **Par3: Well depth e/kT, dimensionless, positive values are attractive** **Par4: Well width, multiples of diameters**
Sticky hard spheres	**no reference given in NIST macros…** **Par1: Radius [A]** **Par2: Vol. fraction** **Par3: Perturbation parameter (0.1)** **Par4: Stickiness, tau**
Hayer Penfold MSA	**no reference given in NIST macros…** **Par1: Radius [A]** **Par2: Charges** **Par3: Volume fraction**

Par4: Temperature in Kelvin
Par5: Monovalent salt concentration (M)
Par6: dielectric constant of solvent

Important comment for Core-shell and Core shell cylinder (and Unified tube)

The volume definition for Core-shell objects is matter of discussion. Heated at times and I suspect that the appropriate answer depends on the case when and how the FF is used. Therefore from version 2.26 Irena macros include option which needs to be set – both Core shell and Core shell cylinder will share common parameter (this parameter is global for all cases of calls to core shell form factors or their volumes) of volume definition.
The options are: whole particle, core, and shell.

Note: Unified tube is using as volume the volume of shell. It is how it is defined at this time and it is meant for cases like Carbon nanotubes, when this is appropriate. To match with core shell cylinder us “shell” as volume of particle.

List and graphs of each form factor included

2.1. Spheroid

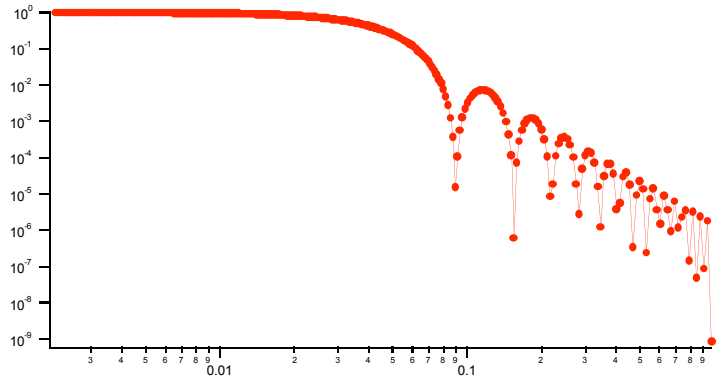
uses sphere form factor for aspect ratio between 0.99 and 1.01:

F^2 = 3/(QR^3))*(sin(QR)-(QR*cos(QR))

volume : V=((4/3)*pi*radius^3)

This calculation approximates integral ove R as rectangle (compare with Integrated spheroid).

graph for R = 50A



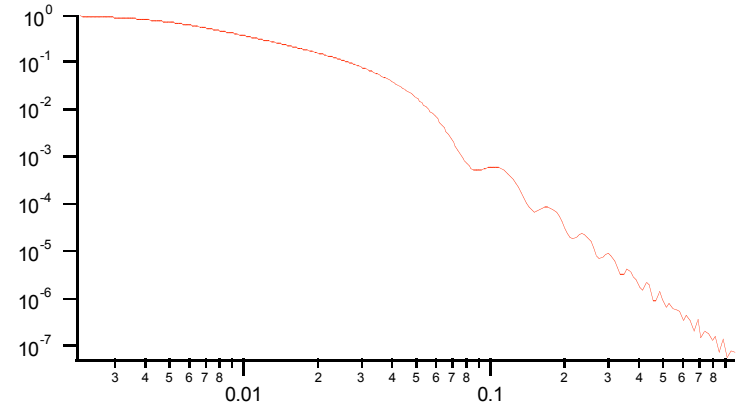
For aspect rations smaller than 0.99 and larger than 1.01 uses standard form factor for spheroid:

F = Integral of (3/(QR^3))*(sin(QR)-(QR*cos(QR)))

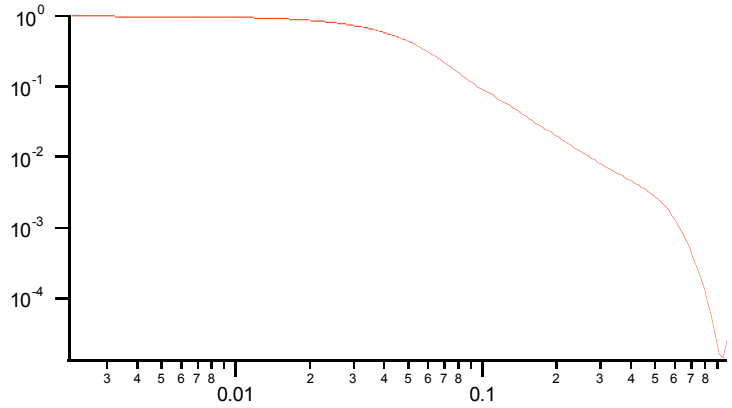
where QR=Qvalue*radius*sqrt(1+(((AR^2)-1)*CosTh^2))

over of CosTh = 0 to 1. This is numerically calculated using 50 points (step in CosTh = 0.02).
Following graphs are examples:

AR = 10



AR=0.1



****
****
****

2.2. Integrated Spheroid

same code as in the spheroid, but in this case the code integrates over the width of the R bin.
Note, the bin star and end points are calcualted linearly (even for log-binned data) as half way distance:

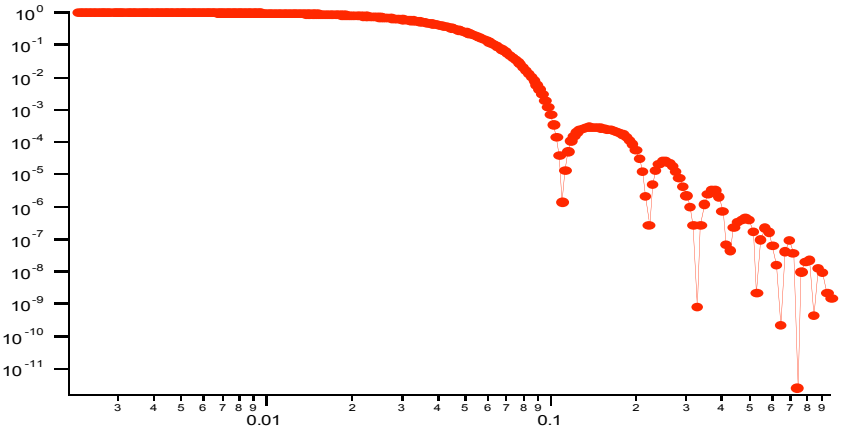
$R_{start} = (R_n + R_{n-1})/2$

$R_{end} = (R_n + R_{n+1})/2$

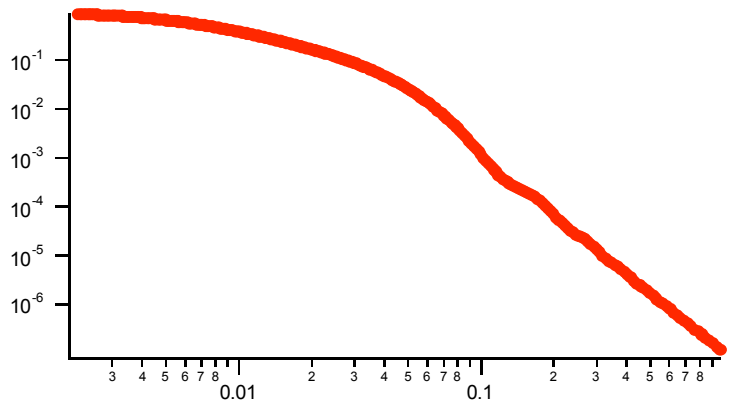
Uses adaptive steps to integrate essel function oscillations of the form factor over the width of the bin in R - note, the averaging is done including the volume of particles involved. This code is quite convoluted and time consuming. Its only reasonable use is for cases with wide bins in radius (R), when this removes some of the bessel function oscillations.

Examples with R width 40A, average size 50A (that means R varies from 30 to 70A). Note that the bessel function oscillations are somewho smooth out. With wider bins in R these oscillations may disappear all together.

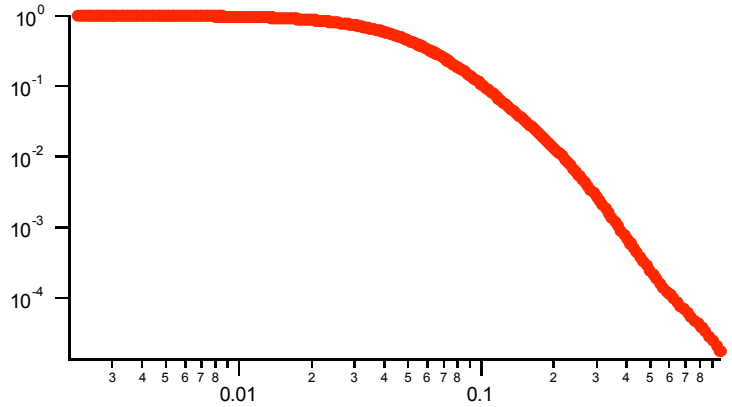
AR = 1 (sphere)



AR=10 (Spheroid)



AR=0.1 (spheroid)



****
****
****

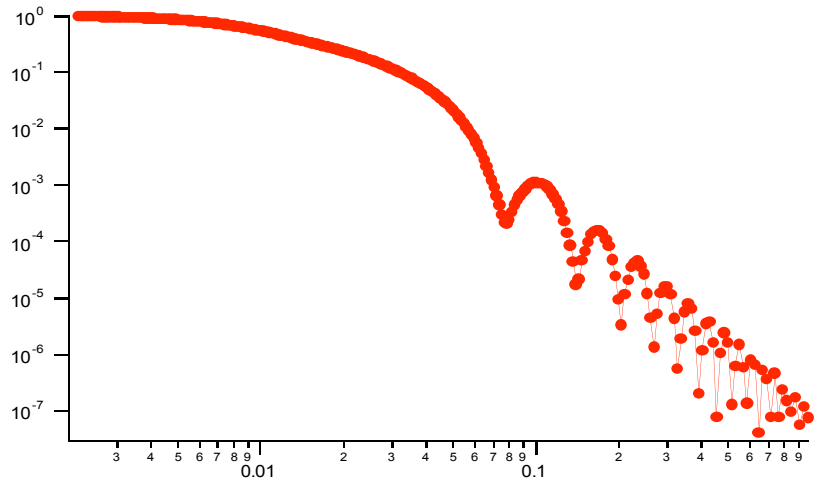
2.3. Cylinder & cylinderAR

The code uses the following code to calculate form factor for cylinder. Note, that also this code is doing the same integration as integrated spheroid above (see 2).

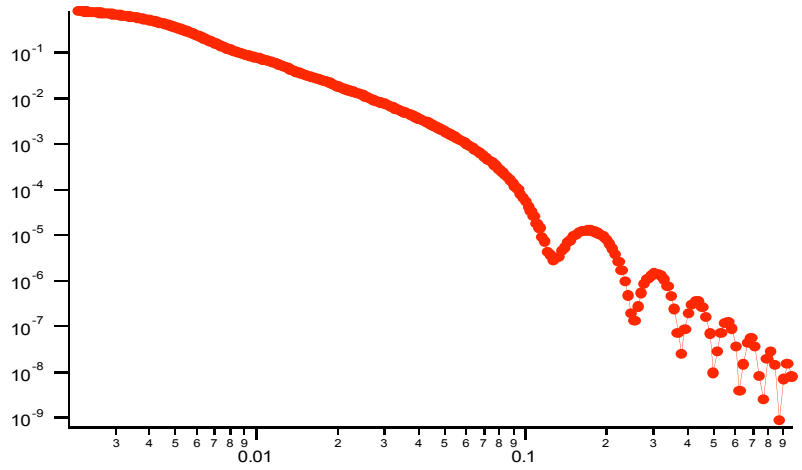
Form factor = integral over (Ft) for Alpha = 0 to pi/2, Ft is below:

LargeBes=sin(0.5*Qvalue*length*Cos(Alpha)) / LargeBesArg
SmallBessDivided=BessJ(1, Qvalue*radius*Sin(Alpha))/Qvalue*radius*Sin(Alpha)
Ft = LargeBes*SmallBessDivided

Examples
Cylinder with length 500A and radius 50A.



Disk (cylinder) with radius 500A and length 50A.



****
****
****

2.4. Algebraic_Globules

Note, that for this model 1/3 <= Aspect ratio <=3 is appropriate

This is form factor created by Andrew Allen, it is a formula, which satisfies the basic form factor requirements:
1. Terminate at 1 at small q
2. Place Gunier region in the right place
3. Have the right terminal slope

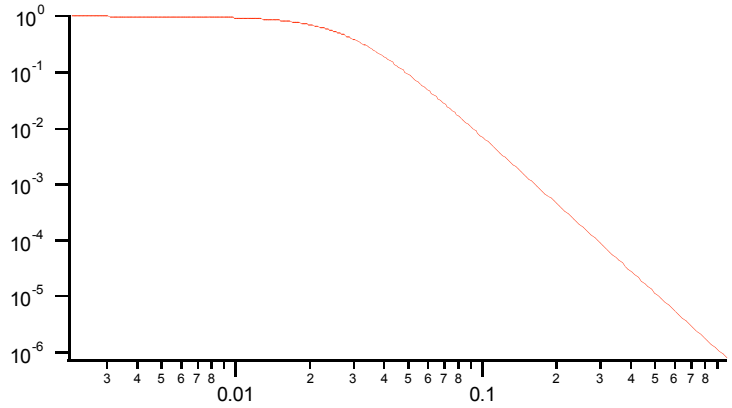
This is the code

```
if(AspectRatio<0.99)
        sqqt = sqrt(1-AspectRatio^2)
        argument = (2 - AspectRatio^2 + 2*sqqt)/(AspectRatio^2)
        surchi = (1 + AspectRatio^2 * ln(argument) / (2*sqqt)) / (2 * AspectRatio)
elseif(AspectRatio>1.01)
        sqqt = sqrt(AspectRatio^2 - 1)
```

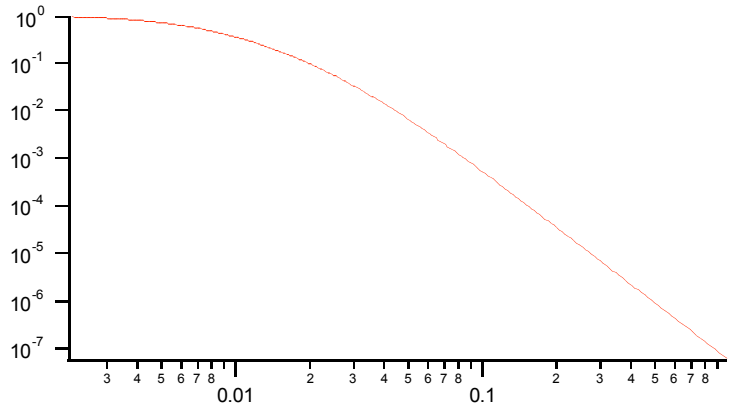
Jan Ilavsky

```
            argument = sqqt / AspectRatio
            surchi = (1 + AspectRatio^2 * asin(argument) / (sqqt)) / (2 * AspectRatio)
else
            //AspectRatio==1
            surchi = 1
endif
QR = currentR * Q_vector
bott = 1 + QR^2 * (2 + AspectRatio^2)/15 + 2 * AspectRatio * QR^4 / (9 * surchi)
F^2 = 1  /  bott
```

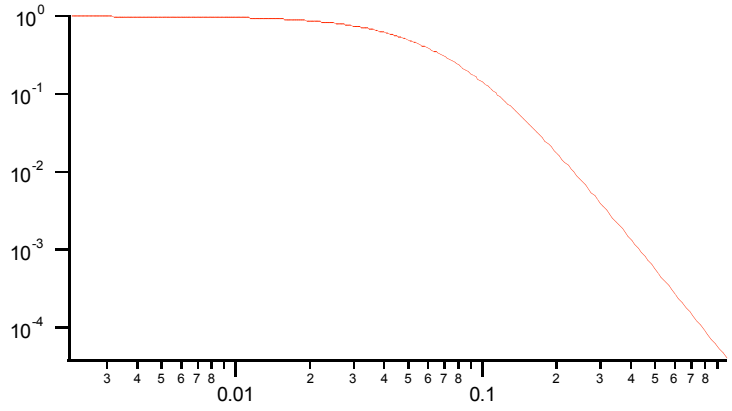
Following are examples
AR=1, R=50A



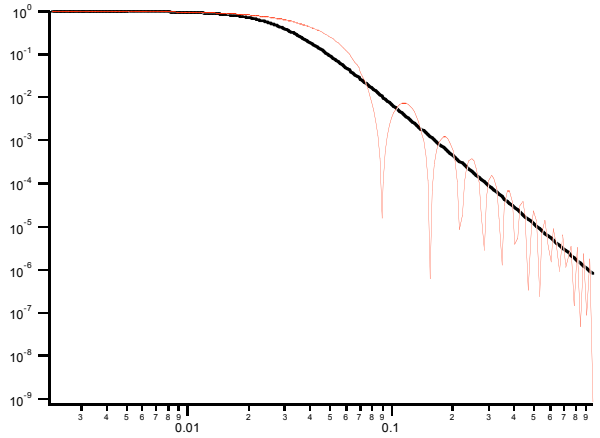
AR=10, R=50A



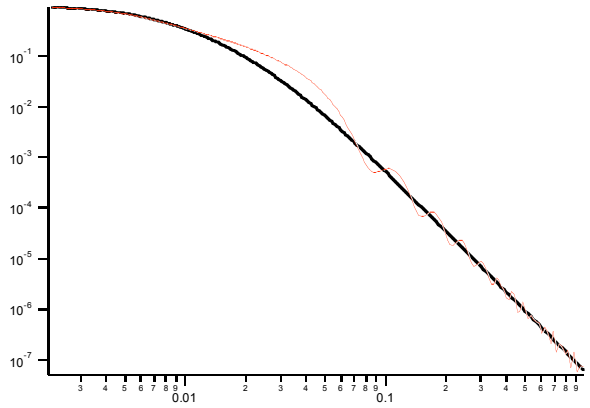
AR=0.1, R=50A



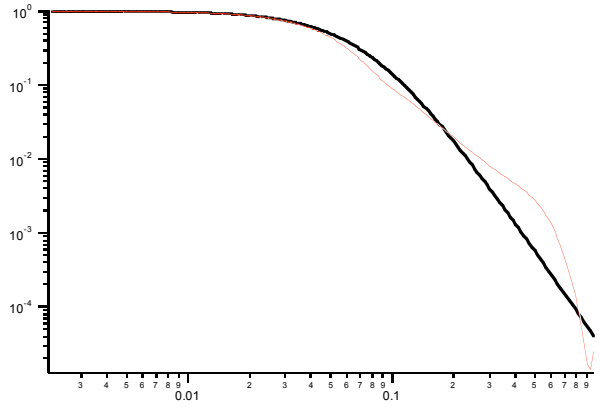
Comparison of Algebraic globule with spheroid:
AR=1



AR=10



AR=0.1



****
****
****

Jan Ilavsky

2.5. Algebraic_Rods

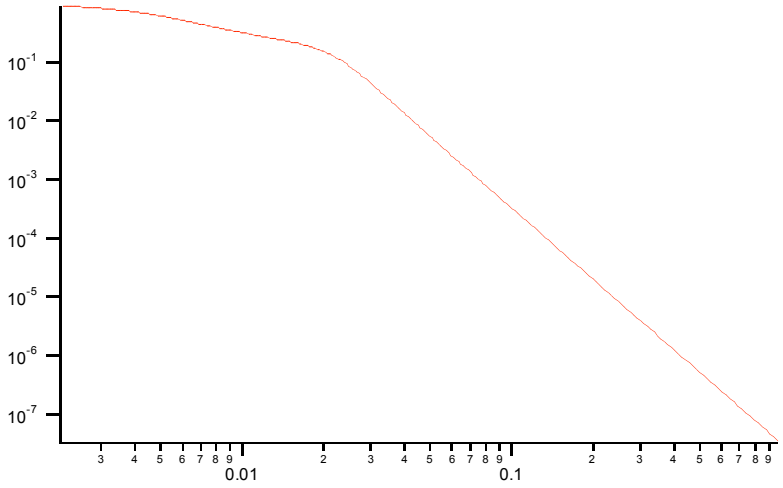
Note, that this formula is valid for Aspect ratio >=10

Another formula Andrew Allen...

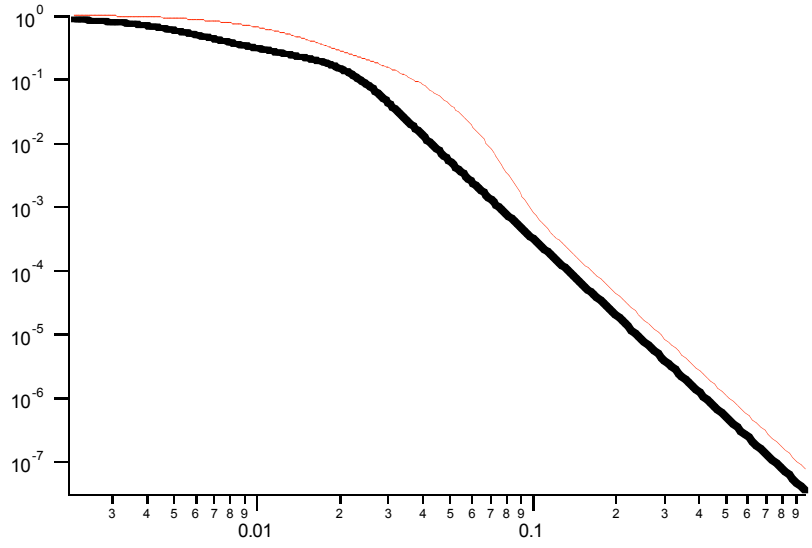
This is the code:

```
QR = R * Q
QH = Q * AspectRatio * R
topp = 1 + 2*pi*QH^3 * QR/(9 * (4 + QR^2)) + (QH^3 * QR^4)/8
bott = 1 + QH^2 * (1 + QH^2 * QR)/9 + (QH^4 * QR^7)/16
F^2 = topp/bott
```

Examples
AR10



Comparison with cylinder AR = 10



Jan Ilavsky

****
****
****

2.6. Algebraic_Disks

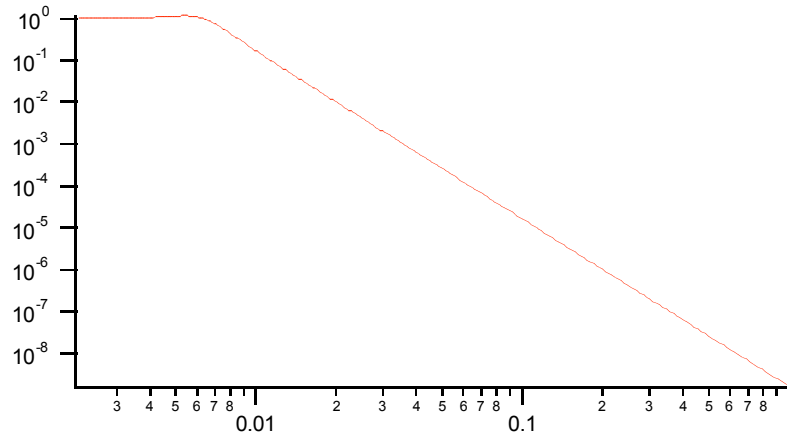
Note, that this formula is valid for Aspect ratio <=0.1

Another formula Andrew Allen...

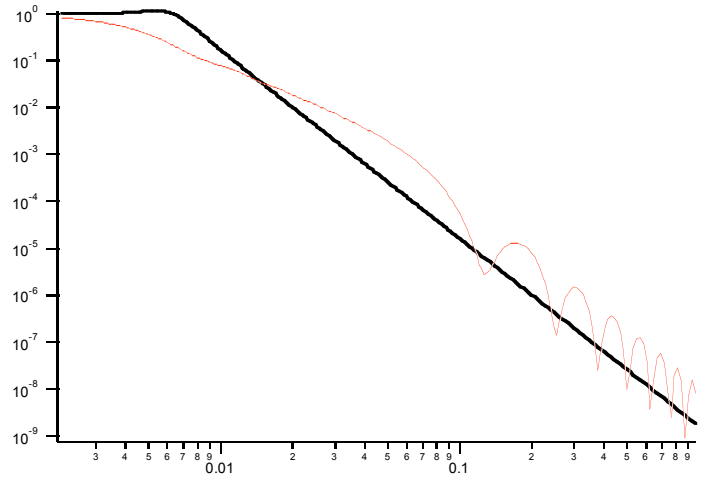
This is the code:

```
QH = Q * R
QRd = Q *  R * AspectRatio
topp = 1 + QRd^3 / (3 + QH^2) + (QH^2 * QRd / 3)^2
bott = 1 + QRd^2 * (1 + QH * QRd^2)/16 + (QH^3 * QRd^2 / 3)^2
F^2 = topp/bott
```

Examples
Radius 500A, AR = 0.1



Comparison with cylinder AR 0.1



Jan Ilavsky

****
****
****

2.7. Unified_Sphere

This is formula from Unified fit model by Greg Beaucage (see Unified tool and documentation for it). The parameters are calculated from the code in the manual for each different shape. Specific formulas for these shapes were provided by Dale Schaefer...

This is the code:

```
G1=1
P1=4
Rg1=sqrt(3/5)*radius
B1=1.62*G1/Rg1^4
QstarVector=qvalue/(erf(qvalue*Rg1/sqrt(6)))^3
F^2 = G1*exp(-qvalue^2*Rg1^2/3)+(B1/QstarVector^P1)
```

Example for R=50A compared with the spheroid with aspect ratio =1

****
****
****

2.8. Unified_Rod & Unified_rodAR

This is formula from Unified fit model by Greg Beaucage (see Unified tool and documentation for it). The parameters are calculated from the code in the manual for each different shape. Specific formulas for these shapes were provided by Dale Schaefer...

This is the code:

```
G2 =1
Rg2=sqrt(Radius^2/2+Length^2/12)
B2=G2*pi/length
P2=1
Rg1=sqrt(3)*Radius/2
RgCO2=Rg1
G1=2*G2*Radius/(3*Length)
B1=4*G2*(Length+Radius)/(Radius^3*Length^2)
P1=4
QstarVector=qvalue/(erf(qvalue*Rg2/sqrt(6)))^3
A=G2*exp(-qvalue^2*Rg2^2/3)+(B2/QstarVector^P2) * exp(-RGCO2^2 * qvalue^2/3)
QstarVector=qvalue/(erf(qvalue*Rg1/sqrt(6)))^3
F^2 = A + G1*exp(-qvalue^2*Rg1^2/3)+(B1/QstarVector^P1)
```

Example for R=50A and length 500A compared with the cylinder

****
****
****

2.9. Unified_Disk

This is formula from Unified fit model by Greg Beaucage (see Unified tool and documentation for it). The parameters are calculated from the code in the manual for each different shape. Specific formulas for these shapes were provided by Dale Schaefer...

This is the code:

```
G2=1
Rg2=sqrt(Radius^2/2+thickness^2/12)
B2=G2*2/(radius^2)//dws guess
P2=2
Rg1=sqrt(3)*thickness/2// Kratky and glatter = Thickness/2
RgCO2=1.1*Rg1
G1=2*G2*thickness^2/(3*radius^2)
B1=4*G2*(thickness+Radius)/(Radius^3*thickness^2)//same as rod
P1=4
QstarVector=Q/(erf(Q*Rg2/sqrt(6)))^3
A=G2*exp(-Q^2*Rg2^2/3)+(B2/QstarVector^P2) * exp(-RGCO2^2 * Q^2/3)
QstarVector=Q/(erf(Q*Rg1/sqrt(6)))^3
F^2 = A + G1*exp(-Q^2*Rg1^2/3)+(B1/QstarVector^P1)
```

Example for R=250A and thickness 10A compared with the cylinder

****
****
****

2.10. CoreShell

Note, this form factor calculation also includes integration over the width of bin in radii (same as integrated spheroid and cylinder).

Note: Input contrast is delta-rho-squared of core to surrounding (solvent/matrix)! this is very important to keep in mind.

Note, that there is volume definition choice you need to do: Whole particle, core, or shell, as appropriate for given problem. This volume definition is used for all volume calculations for this particle. It is global parameter for all core shell cylinder or core shell calls in the WHOLE EXPERIMENT….

Code (heavily simplified!):

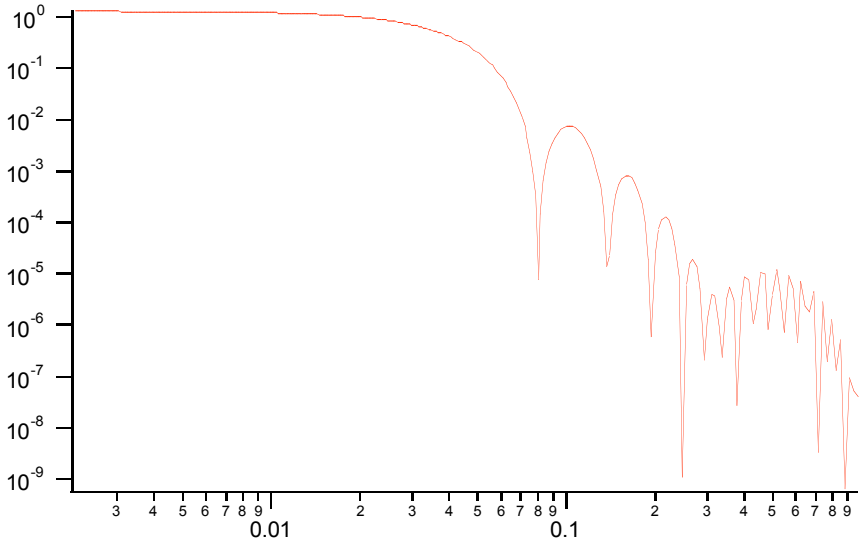
```
Contrast = CoreRho - ShellRho
//core
Result1=(3/(Q*R)^3)*(sin(Q*R)-(Q*R*cos(Q*R)))^2  * Contrast * (4/3 * pi * R^3)

//Now add the shell (skin) , thickness Rshell
r = R+Rshell
Contrast = ShellRho - SolventRho
Result2 = (3/(Q*r)^3)*(sin(Q*r)-(Q*r*cos(Q*r)))^2 * Contrast * (4/3 * pi * r^3)

//summ them together and normalize by the total particle volume
F^2 =( result1 + result2 )^2 / Volume
```

Volume definition depends on the setting of above discussed global parameter and is either:
Whole particle volume = 4/3 * pi * (R+r)^3
Core volume = 4/3 * pi * R^3
Shell volume = 4/3 * pi * r^3

Make sure your choice is appropriate

Note, that to my surprise these calculations (copied from NIST Form factors) do not normalize correctly to 1 at low q. The reason is that the weighting is done by volume and contrast. I'll need to look into this again and in detail...

Example, Radius 50A, skin thickness 10A, contrast ratio 0.6



****
****
****

2.11. Core shell cylinder

Note, this form factor calculation also includes integration over the width of bin in radii (same as integrated spheroid and cylinder).

This code has been developed some time ago and I am not sure about it's function...

Code which is being used is direct copy of NIST Core shell cylinder.

Note, that there is volume definition choice you need to do: Whole particle, core, or shell, as appropriate for given problem. This volume definition is used for all volume calculations for this particle. It is global parameter for all core shell cylinder or core shell calls in the WHOLE EXPERIMENT….

Volume definition depends on the setting of above discussed global parameter and is either:
Whole particle volume = pi * (R+r)^2 * (L+2*r)
Core volume = pi * R^2 * L
Shell volume = pi * r^2 * L

****

Jan Ilavsky

****
****

2.12. Fractal Aggregate

This form factor was requested by Dale Schaefer and I cannot very well guarantee its functionality....

code:

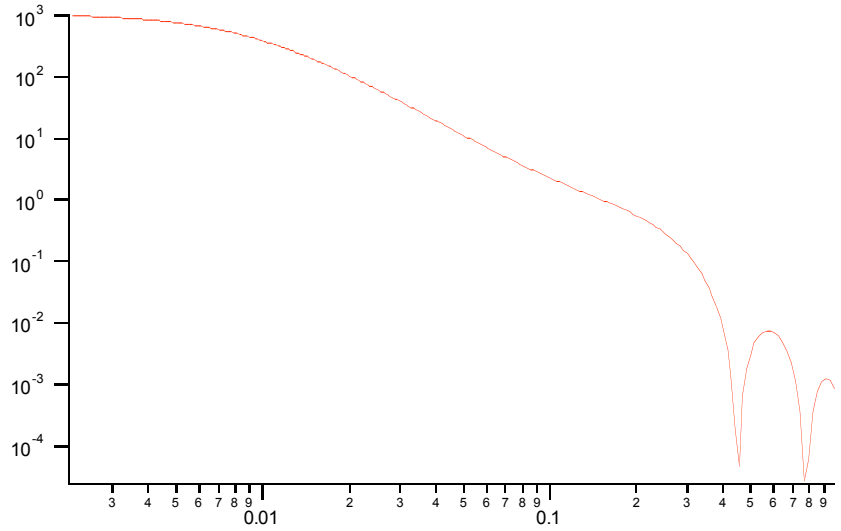
```
f = IR1T_CalcSphereFormFactor(Qw[p],(2*Param1))
//calculates the F(Q,r) part fo formula
//this is same as for sphere of diameter = 2*Param1
//(= radius of primary particle, which is hard sphere)
//fractal part is next
F^2 =f^2 * IR1T_CalculateFractAggSQPoints(Qw[p],currentR,Param1, Param2)
```

where
IR1T_CalculateFractAggSQPoints(Qvalue,R,r0, D) is

```
QR=Qvalue*R
part1=1
part2=(qR*r0/R)^-D
part3=D*(exp(gammln(D-1)))
part5= (1+(qR)^-2)^((D-1)/2)
part4=abs(sin((D-1)*atan(qR)))
 return (part1+part2*part3*part4/part5)
```

Note, that parameters are :
Param1 - radius of primary particle
param2 - fractal dimension of the fractal particles

Example for R=100A, radius of primary particle 10 A and fractal dimension 2.5.



Comment: Note, that this is not scaled correctly at all... I have no idea why - apparently this formula is either wrongly coded or plainly does not behave right.

3. Testing and using Form factors in users own code

To verify that the form factor works for you and to use the form factor if your own functions use following process and functions:

1. Generate Q wave with Qs for which the data are to be calculated
2. Generate intensity wave (will be redimesnioned as necessary, so the only thing is, it should be double precision).
3. Generate distributipon of radii wave - if you want to use single R, create wave with single point
4. decide what you want to calculate:

F^2	powerFct=0
V*F^2	powerFct=1
V^2 * F^2	powerFct=2

5. Run following command:

```
IR1T_GenerateGMatrix(R_FF,Q_wave,R_dist,powerFct,"form factor
       name",param1,param2,param3,param4,param5, "", "")
```

This function will return R_intensity, which is generally matrix with dimensions numpoints(Q_vector) x numpoints(R_dist), if R_dist has 1 point only, returned is wave (vector) as expected and reasonable...
The param1 - param5 are form factor parameters, as desribed in chapter 1, the "" at the end are for user form factor functions (there go the strings with names of user form factor and volume function).
"form factor name" is name from list in chapter 1.

6. Create log-log plot of the data if R_dist has single point. If R_dist has more point, well, you have to pull out the right column of data you need to plot.

Note, that if the IR1T_GenerateGMatrix function returns wave of NaN values if unknown name of form factor is passed in.

Example of code:

```
make/N=100 Q_wave
Q_wave=0.001+p/100
       //will create 100 points wave with values 0.001 to 1) values
Make/O/D R_FF
       //makes some place for form factor
make R_dist
R_dist=50
//or
//make/N=3 R_dist
//R_dist={10,50,100}
       //creates R distribution and sets values
IR1T_GenerateGMatrix(R_FF,Q_wave,R_dist,powerFct,"form factor
       name",param1,param2,param3,param4,param5, "", "")
```

```
       //Note, above lines belong on one line together!
       // replace powerFct with 0, 1,or 2!
// replace "form factor name" with name of form factor you want to use
Display R_FF vs Q_wave
ModifyGraph log=1
```
//creates log-log graph of

Structure factors

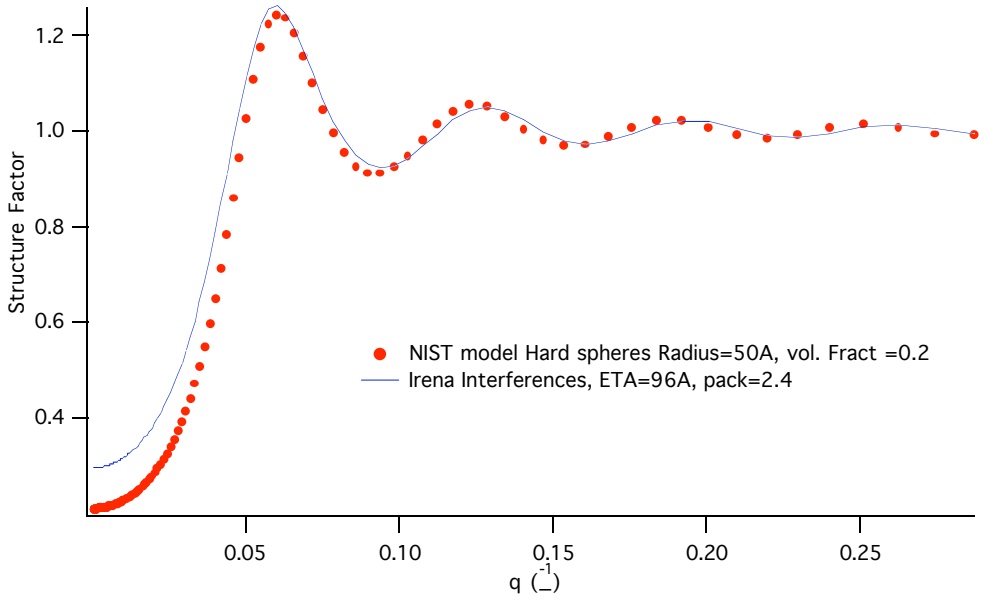
This is list of library of structure factors. These structure factors enable to deal with limited S(Q) effects in Irena package. The functionality is provided by library, which can be called by any other user code. The library provides also GUI for setting the user parameters. In principle, further structure factors can be added if they have less than 5 parameters.

Interferences

This is original structure factor in Irena package. It has been provided as part of Unified fit model by Gregg Beaucage and is listed in his publication: Beaucage, G. (1995). J Appl Crystallogr **28**, 717-728.

$$S(Q) = \frac{1}{1 + k * \frac{3 * (\sin(Q\xi) - Q\xi \cos(Q\xi))}{(Q\xi)^3}}$$

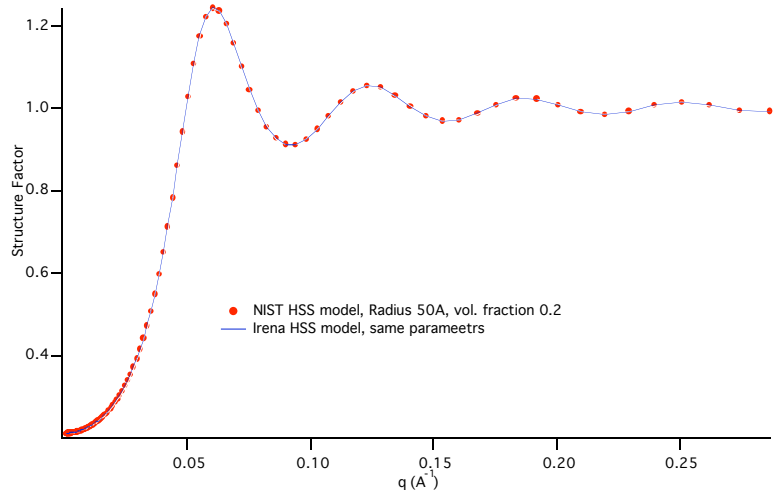
Note, that this model is, for most practical purposes, close to Hard spheres model with different definition of the parameters k ("pack") and ζ ("ETA"). Modeling II extends the capabilities by including three more structure factors using code available from NIST Igor package (ref). Included are now: Hard spheres, Square Well, and Sticky Hard Spheres, which can be used in addition to interferences model above and no structure factor (dilute limit).



Hard spheres

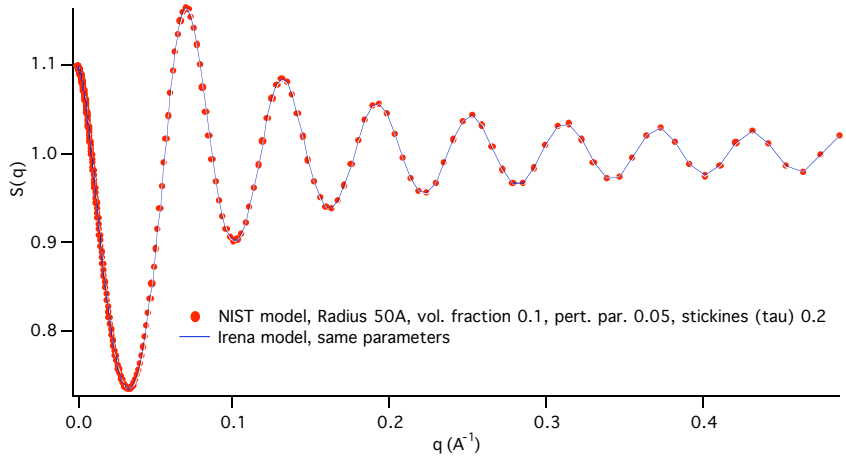
The code for this structure factor has been copied from NIST SAS macros (Kline, S. R. (2006). J Appl Crystallogr **39**, 895-900). Please, give them credit when using this structure factor. (http://www.ncnr.nist.gov/programs/sans/data/data_anal.html)…

This is graph of NIST model and Irena implementation.



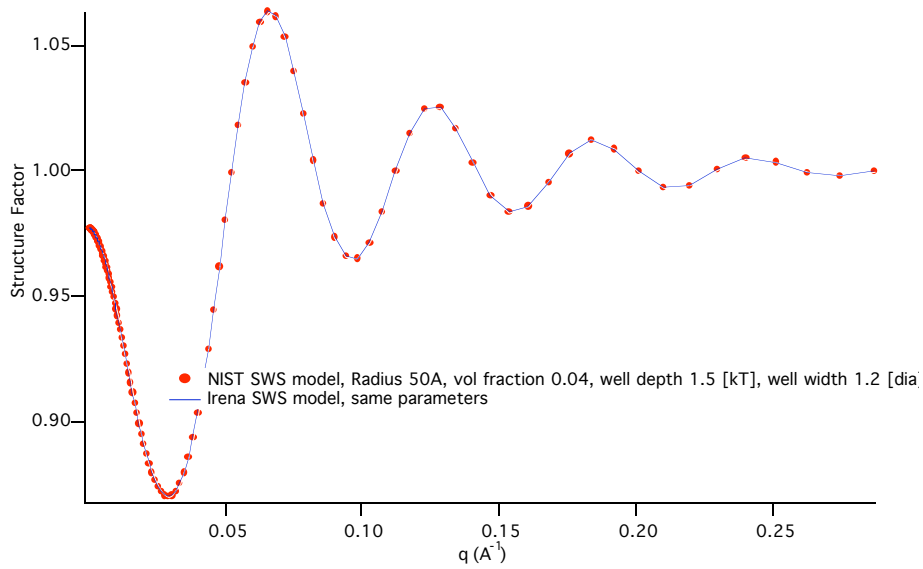
Sticky hard spheres

The code for this structure factor has been copied from NIST SAS macros (Kline, S. R. (2006). J Appl Crystallogr **39**, 895-900). Please, give them credit when using this structure factor. (http://www.ncnr.nist.gov/programs/sans/data/data_anal.html)…



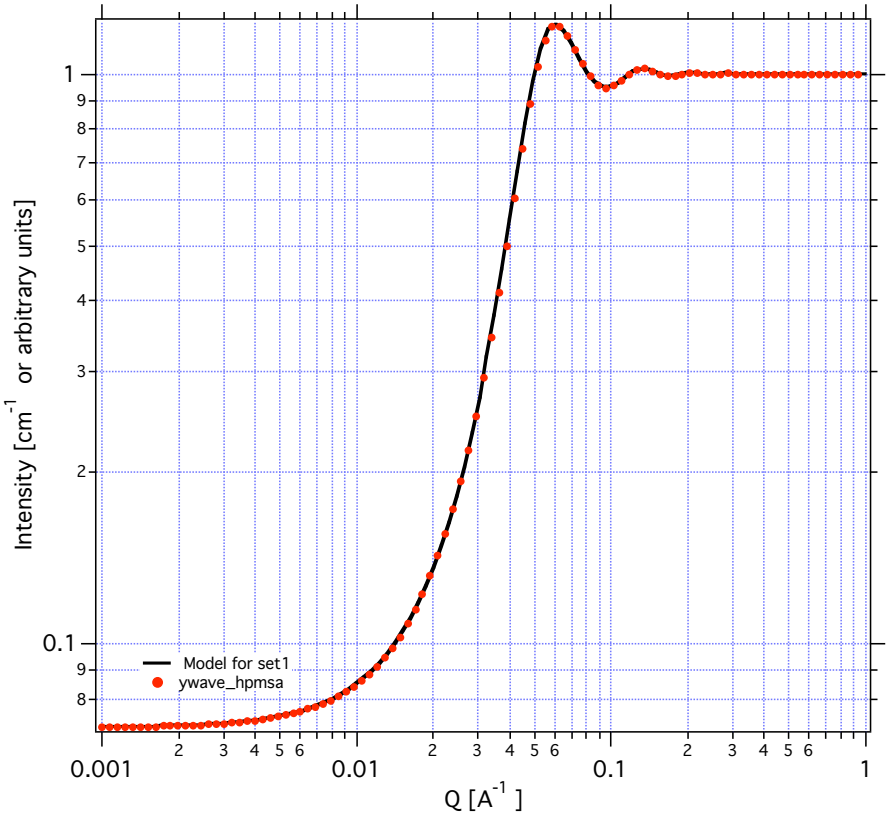
Square well

The code for this structure factor has been copied from NIST SAS macros (Kline, S. R. (2006). J Appl Crystallogr **39**, 895-900). Please, give them credit when using this structure factor. (http://www.ncnr.nist.gov/programs/sans/data/data_anal.html)…



Hayter Penfold MSA model

The code for this structure factor has been copied from NIST SAS macros (Kline, S. R. (2006). For any details on the use of these, please see the original code and description from NIST data analysis package (http://www.ncnr.nist.gov/programs/sans/data/data_anal.html)... Please, give them credit when using this structure factor.



This is graph fro standard NIST set of parameters for both Irena package (black line) and NIST package

(red dots). Both assume ONLY structure factor (Form factor is set to 1). The parameters were:
Diameter (A) 41.5 NOTE: Irena uses here radius, which is converted to diameter inside the structure factor. This is to keep consistency with other structure factors.
Charge 19
Volume Fraction 0.0192
Temperature(K) 298
monovalent salt conc. (M) 0
dielectric constant of solvent 78
Units are mentioned in the help for each filed on the Structure factor panel (you may have to enable help on Mac, it is shown always on PC in the bottom left corner of the Igor window).
Important note: this is comment from original NIST code….
// *** NOTE * THIS CALCULATION REQUIRES THAT THE NUMBER OF***
// Q-VALUES AT WHICH THE S(Q) IS CALCULATED BE
// A POWER OF 2
//!!!!! this is at this time NOT enforced in Irena implementation...
// I am not sure if this is really problem or not.
// How do I find out? Users need to test this for me and if necessary, I need to try it out.
// in my testing there was NO problem with the results when the number of q pointds was arbitrary number of points...

Calling the library and use

Users can use built in library in their own code using following calls:

1. initialize by calling: IR2S_InitStructureFactors()
this is where the list of known structure factors is:
SVAR ListOfStructureFactors=root:Packages:StructureFactorCalc:ListOfStructureFactors

2. use by calling:
IR2S_CalcStructureFactor(SFname,Qvalue,Param1,Param2,Param3,Param4,Param5,Param6)

I(Q) = I(Q, dilute limit) * IR2S_CalcStructureFactor(SFname,Qvalue,Param1,Param2,Param3,Param4,Param5,Param6)

//Dilute system;Interferences;HardSpheres;SquareWell;StickyHardSpheres;HayterPenfoldMSA

3. Get panel by calling:
IR2S_MakeSFParamPanel(TitleStr,SFStr,P1Str,FitP1Str,LowP1Str,HighP1Str,P2Str,FitP2Str,LowP2Str,HighP2Str,P3Str,FitP3Str,LowP3Str,HighP3Str,P4Str,FitP4Str,LowP4Str,HighP4Str,P5Str,FitP5Str,LowP5Str,HighP5Str, P6Str,FitP6Str,LowP6Str,HighP6Str,SFUserSFformula)
to disallow fitting of paramters, simply set FitP1Str="" etc.
then do not have to set low and high limits ...

Structure factors package...
IR2_OldInterferences this is roughly hard spheres (close to Percus-Yevick model, not exactly), the ETA = 2* radius and Phi = 8 * vol. fraction for PC model.
IR2_HardSphereStruct this is Percus-Yevick model
IR2_StickyHS_Struct this is sticky hard spheres
IR2_SquareWellStruct this is Square well
IR2_HayterPenfoldMSA this is HayterPenfoldMSA