

SPPr Package (Version 1.0)

Baptiste Auguie

February 12, 2010

```
Package: SPPr
Type: Package
Title: Surface Plasmon Polaritons excitation in (multilayered) Kretschmann geometry
Version: 1.0
Date: 2008-01-13
Author: Baptiste Auguie
Maintainer: baptiste <ba208@exeter.ac.uk>
Description: Provides a modelling of multilayer reflection (transmission)
             coefficient, convenient functions to reshape experimental data, plot it against
             the model and fit.
License: GPL (>=2)
URL: http://r-forge.r-project.org/projects/photronics/
LazyLoad: yes
Repository: R-Forge
Repository/R-Forge/Project: photronics
Repository/R-Forge/Revision: 482
Date/Publication: 2010-02-12 08:46:43
```

1 Introduction

This document provides a tutorial example using the **SPPr** package to model and fit surface plasmon angle scans. Below is a list of the available functions.

```
> library(SPPr)
> ls("package:SPPr")

[1] "angleScan"      "criticalEdge"
[3] "data2norm"      "ext2int"
[5] "fitModel"       "objF"
[7] "pIni"           "plotData"
[9] "plotModel"      "reflModel"
[11] "spp.model"      "transmission"
[13] "updateParameters"
```

2 Modelling using recursive Fresnel coefficients

The physical picture describing the excitation of surface plasmons on a thin metallic film in the Kretschmann geometry is well understood^[1] yet quite subtle^[2]. A deceptively simple and rigorous description of the optical response of the system can be done using the Fresnel coefficients, as detailed below. It is however rather difficult to extract the physical meaning of the surface plasmon polariton from this formulation. Raether^[1] points out that the poles in the complex reflectivity of the system describe the existence of electromagnetic eigenmodes of the system. In physical terms, we are asking for a solution of the Maxwell equations in the form of a bound surface mode. At a simple (unperturbed)

metal/dielectric interface, applying the standard boundary conditions for the macroscopic EM field leads to an expression of the propagation constant of the surface mode in the form,

$$k_{\text{SPP}} = k_0 \sqrt{\frac{\varepsilon_m \varepsilon_d}{\varepsilon_m + \varepsilon_d}},$$

the excitation of SPPs by the incident light will occur when the (tangential) momentum of the light inside the prism matches the momentum of this mode,

$$n \cdot \sin \theta_{\text{int}} = \sqrt{\frac{\varepsilon_m \varepsilon_d}{\varepsilon_m + \varepsilon_d}}.$$

The situation is however slightly more complicated as the dispersion of the mode is modified by the presence of the second interface of the thin metallic film. In physical terms, the process of excitation

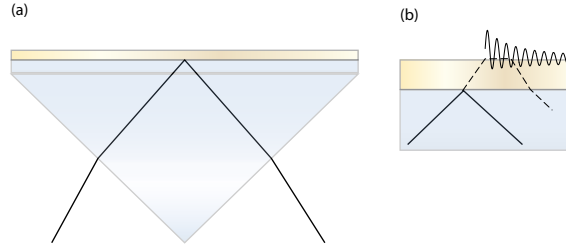


Figure 1: Schematic description of the Kretschmann configuration.

of SPPs can be understood as follows: First, the incident light gains momentum inside the prism ($k = nk_0$). The reflected light that is detected outside the prism can come from two reflection channels,

1. upon total internal reflection (for angles above the critical edge), part of the incident light is reflected at the metal/prism interface
2. as in frustrated total internal reflection, there is a probability that the incident light couples to the surface plasmon polariton on the other side of the thin metallic film. The light converted into this surface mode can decay in two forms: non-radiative (joule heating of the film), or radiative. It is clear that if the light inside the prism had a momentum commensurate with the SPP mode, the reverse conversion from SPP to light can occur: the surface plasmon mode re-radiates light in the prism.

It is the interference of these two ‘reflection’ channels that is measured by the detector (therefore, their relative phase and amplitude matter). When the thickness of the film is such that the radiative and non-radiative losses are equal, 100% absorption can occur. The energy is then completely converted into heat inside the film. The existence of an optimum thickness stems from the balance of two opposite constraints: a thick film will allow very little overlap between the incident field and the SPP mode; a film too thin will see a higher amplitude of the second channel (SPP re-radiating light).

2.1 Fresnel coefficients

Fresnel coefficients for a single interface read,

$$r_{01}^p = \frac{k_{z1}/\varepsilon_1 - k_{z2}/\varepsilon_2}{k_{z1}/\varepsilon_1 + k_{z2}/\varepsilon_2}, \quad r_{01}^s = \frac{k_{z1} - k_{z2}}{k_{z1} + k_{z2}} \quad (1)$$

$$t_{01}^p = \frac{2nk_{z1}}{nk_{z1} + k_{z2}}, \quad t_{01}^s = \frac{2k_{z1}}{k_{z1} + k_{z2}} \quad (2)$$

Note that,

$$t_{ij} = -t_{ji},$$

and from the conservation of energy,

$$1 = t_{ij}^* t_{ji} + r_{ij}^* r_{ji}$$

2.2 Reflectivity of a layer

A thin layer will in general support an infinite number of internal reflections. The infinite series of reflected orders can be expressed in the form a geometric sum, leading to a closed form formula as shown below. In “A”, we have an incident plane wave of amplitude A . It can be reflected, $B = r_{01}A$,

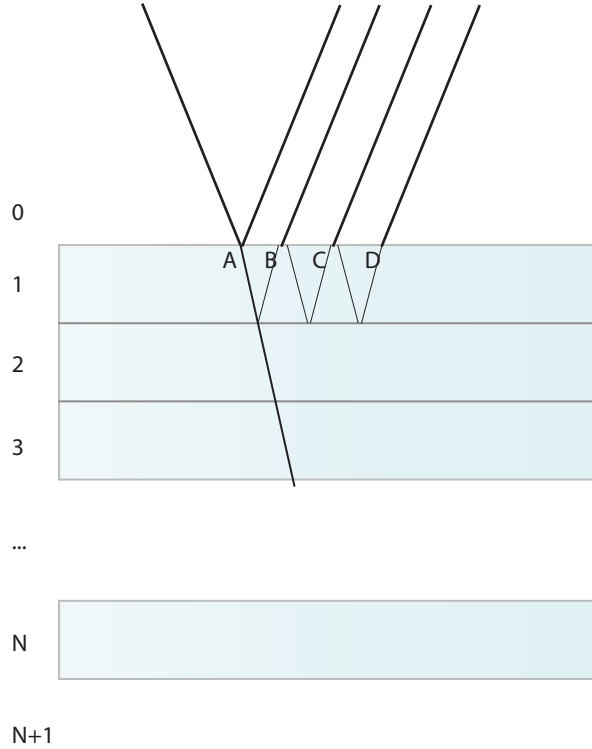


Figure 2: Schematic of a multilayer stack. A few reflected orders are noted A, B, C and D for the first interface.

or transmitted. We now look at each order of reflection inside the slab (“C”, “D”, ...).

Upon transmission, the wave is now $t_{01}A$. Fermat’s principle leads to a phase change $\Delta\phi = k_{z1}d$ when the wave hits the second interface. The reflection coefficient at this interface is r_{12} . The partial wave reflected from this path, call it “C” is therefore, $C = t_{10}t_{01}r_{12} \exp(2ik_{z1}d)A$.

Similarly, we get for “D”,

$$D = t_{10}t_{01}r_{10}r_{12}^2 \exp(4ik_{z1}d)A$$

And, for the j^{th} partial wave,

$$t_{10}t_{01}r_{12}^j r_{10}^{j-1} \exp(2jk_{z1}d)A$$

The wave reflected by the slab is the sum of these partial contributions,

$$r_{\text{slab}}A = B + C + D + \dots = (r_{01} + t_{10}t_{01}r_{12}) \sum_{j=0}^{\infty} r_{12}^j r_{10}^j \exp(2jk_{z1}d)A$$

Let us define $\beta := r_{12}r_{10} \exp(2ik_{z1}d)$. We recognize the desired geometrical sum,

$$r_{\text{slab}} = (r_{01} + t_{10}t_{01}r_{12} \exp(2ik_{z1}d)) \sum_{j=0}^{\infty} \beta^j$$

Now, recalling that the sum of a geometric series of common ratio q is $\frac{1}{1-q}$, we can write,

$$r_{\text{slab}} = \frac{r_{01} + t_{10}t_{01}r_{12} \exp(2ik_{z1}d)}{1 - r_{12}r_{10} \exp(2ik_{z1}d)}$$

Using the relations stated above we finally get,

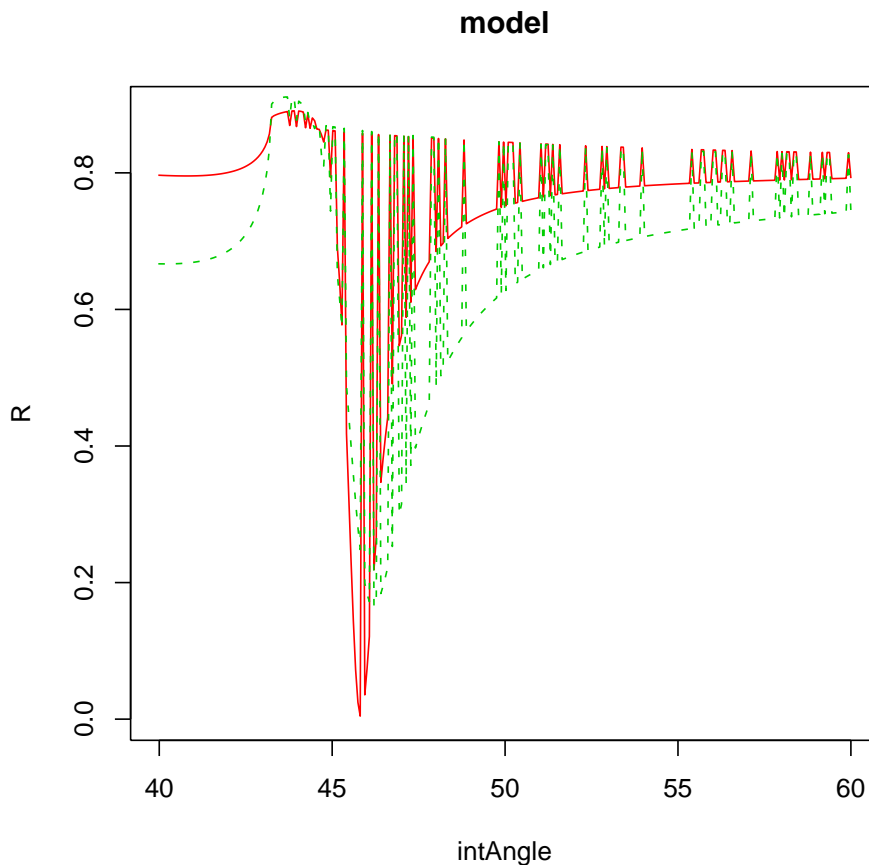
$$r_{\text{slab}} = \frac{r_{01} + r_{12} \exp(2ik_{z1}d)}{1 - r_{12}r_{10} \exp(2ik_{z1}d)}.$$

When N layers are stacked together, the reflection coefficient of the structure can be found by applying recursively the preceding formula for a single layer. This amounts to considering one of the reflection coefficients to be the effective reflection accounting for all the layers behind.

3 Kretschmann angle scan

The following code illustrates how to define the optical system (list of parameters) and plot the reflectivity as a function of incident angle.

```
> library(SPPr) # load the library
> # require(constants) # permittivity data
> # require(baptMisc) # utilities such as clr() to clear the workspace
> # clr()
>
> parameters <- list(epsilon = c(1.46^2, -12+1.2i, 1+0i),
+                       thickness = c(NA, 50, NA),
+                       lambda = 632.8,
+                       k0 = 2*pi/632.8,
+                       anglePrism = pi/3,
+                       zeroShift = 0.0,
+                       normalisation = 23) # this is the normalisation ratio signal/ref without prism
> plotModel(parameters, angle=c(-30, 0), add=F, col=2)
> title("model")
> parameters$thickness[2] <- 35 # change the thickness for fun
> plotModel(parameters, angle=c(-30, 0), add=T, col=3, lty=2) # superpose the result
```



It's easy to define a new function to observe the effect of different parameters. For example,

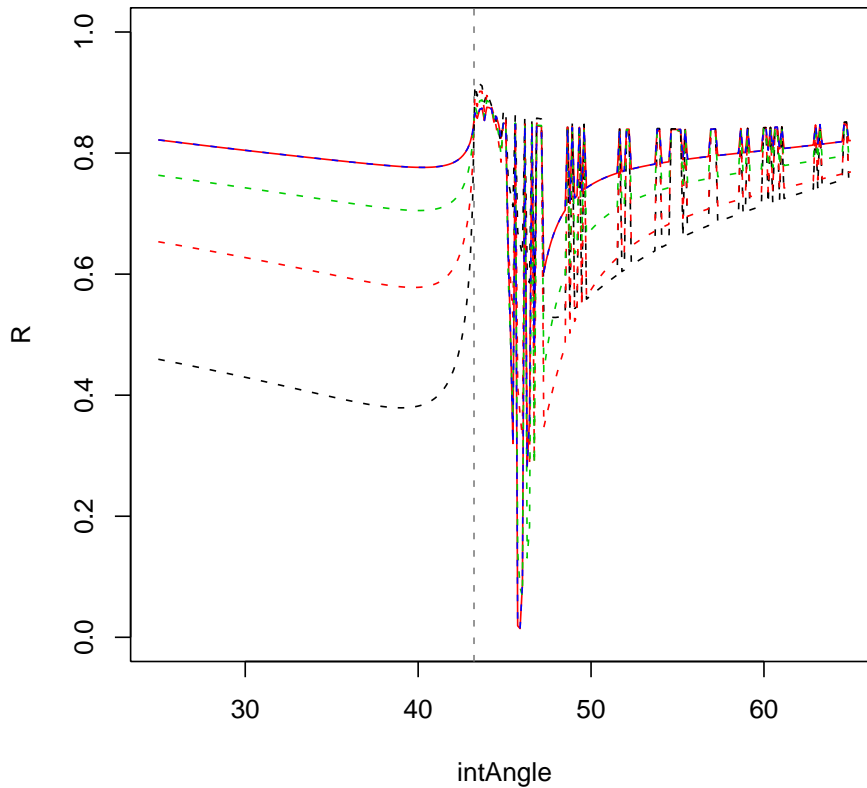
```
> library(SPPPr) # load the library
> # require(constants) # permittivity data
> # require(baptMisc) # utilities such as clr() to clear the workspace
> # clr()
>
> parameters <- list(epsilon = c(1.46^2, -12+1.2i, 1+0i),
+                       thickness = c(NA, 50, NA),
+                       lambda = 632.8,
+                       k0 = 2*pi/632.8,
+                       anglePrism = pi/4,
+                       zeroShift = 0.0,
+                       normalisation = 23) # this is the normalisation ratio signal/ref without prism
> test <- plotModel(parameters, angle=c(-30, 30), add=F, col=2, ylim=c(0, 1))
> title("model with different thicknesses")
> criticalEdge(parameters, col=grey(0.5), lty=2)
> oneThickness <- function(d, quiet=T, angle=c(-30, 30), add=T, ...){
+
+   parameters$thickness[2] <- d
+   result <- plotModel(parameters, angle=angle, quiet=quiet, add=add, ...)$R
+   result
+ }
```

```

> thicknesses <- seq(20, 50, by=10)
> results <- sapply(thicknesses, oneThickness)
> matlines(test$intAngle, results, lty=2)

```

model with different thicknesses



The data should consist of three columns: external angle, signal, reference. This data can be loaded using `read.table` into a `data.frame`. The names of the columns must be **angle** **signal** **reference** as in the example. Use `head()` to check the imported data against the following example data set.

```

> data(scanSPP) # example of measured data.frame:
> # external angle, signal, reference, see ?read.table
> # typically, myData <- read.table("myFile.txt")
> # possibly needs some calculations, such as:
> # myData <- within(myData, angle <- - angle / 2 )
>
> names(scanSPP) <- c("angle", "signal", "reference")
> head(scanSPP) # first few lines, note the column names

```

	angle	signal	reference
1	-50.00	0.008915012	0.0004875688
2	-49.95	0.008934081	0.0004904310
3	-49.90	0.008927400	0.0004875680
4	-49.85	0.008874000	0.0004836340
5	-49.80	0.008904521	0.0004870910
6	-49.75	0.008929311	0.0004891000

```

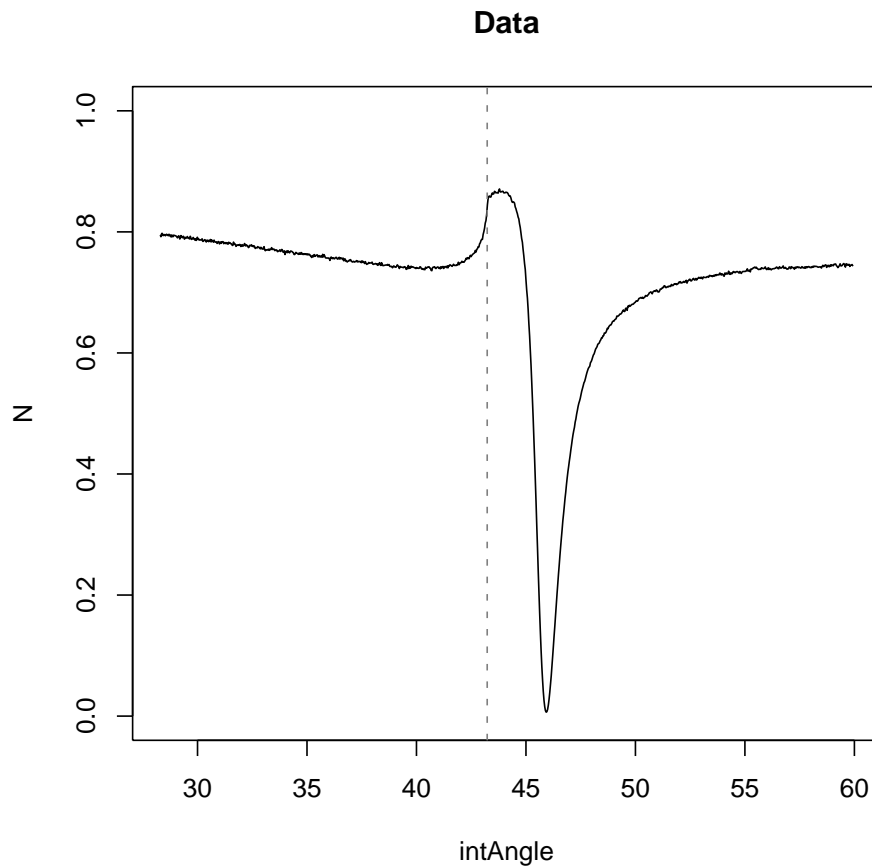
> dataNorm <- data2norm(scanSPP,parameters) # this function uses the parameters
> # to calculate the internal angle
> # and normalized intensity
> head(dataNorm)

      intAngle  intAngle_r extAngle extAngle_r
1 13.35276+0i 0.2330496+0i  -50.00 -0.8726646
2 13.37863+0i 0.2335011+0i  -49.95 -0.8717920
3 13.40452+0i 0.2339529+0i  -49.90 -0.8709193
4 13.43042+0i 0.2344051+0i  -49.85 -0.8700466
5 13.45635+0i 0.2348576+0i  -49.80 -0.8691740
6 13.48230+0i 0.2353105+0i  -49.75 -0.8683013

      kx      S      R      N
1 0.003347929+0i 0.008915012 0.0004875688 0.7949836
2 0.003354297+0i 0.008934081 0.0004904310 0.7920346
3 0.003360669+0i 0.008927400 0.0004875680 0.7960896
4 0.003367045+0i 0.008874000 0.0004836340 0.7977646
5 0.003373425+0i 0.008904521 0.0004870910 0.7948270
6 0.003379810+0i 0.008929311 0.0004891000 0.7937659

> parameters <- list(epsilon = c(1.46^2, -12+1.2i, 1+0i),
+                      thickness = c(NA, 50, NA),
+                      lambda = 632.8,
+                      k0 = 2*pi/632.8,
+                      anglePrism = pi/3,
+                      zeroShift = 0.0,
+                      normalisation = 23)
> parameters$zeroShift <- -0.1 # manual offset if the zero was not exactly right
> dataNorm <- data2norm(scanSPP,parameters)
> plotData(dataNorm, internal=TRUE, new=TRUE, col=1, ylim=c(0, 1))
> criticalEdge(parameters, col=grey(0.5), lty=2)
> title("Data")

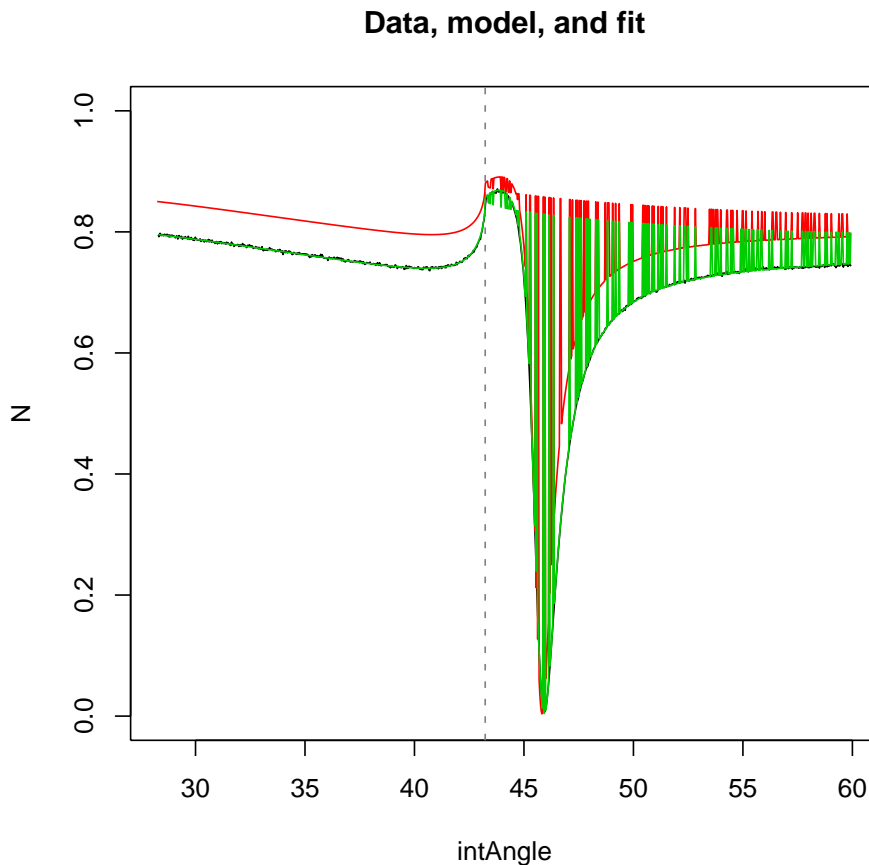
```



```

> plotData(dataNorm, internal=TRUE, new=TRUE, col=1, ylim=c(0, 1))
> criticalEdge(parameters, col=grey(0.5), lty=2)
> title("Data, model, and fit")
> plotModel(parameters, angle=dataNorm$extAngle, quiet=F, add=TRUE,col=2)
> layer2fit <- 1 # specify which layer is the one to fit
> p.ini <- pIni(parameters, layer2fit)
> # result <- optim(p.ini, objF) # call the optimization routine
> # commented to make test run faster
> result <- list(par=c(-11.868123 , 1.279103, 45.340602 , 1.027804))
> yFin <- fitModel(result$par)
> yErr<-abs(yFin-dataNorm$N)
> lines(dataNorm$intAngle, yFin, col=3)

```

Fitting multiple layers requires some thought. The general idea is to define an objective function (get inspiration from `objF`), and call `optim` with a suitable first guess. Evidently, the objective function must depend on the set of parameters that one wishes to optimize.

4 Misc. questions

Exporting the data to a text file.

The first step is to create a matrix or `data.frame` with the data you want to export. The functions `cbind`, `names`, and `as.data.frame` can be helpful. `str` and `head` can help check the form of the data. Once this is done, the function `export2excel` or `write.table` can be used. See `?write.table` for options such as the character separator, *etc.*

The data is plotted with the opposite angle range.

The external angle is not output in the same way in all the kits. The following line will convert the angle θ_{ext} to $-\theta_{\text{ext}}/2$: `my.data <- within(my.data, angle <- -angle/2)`

References

- [1] H.RAETHER. *Surface plasmons on smooth and rough surfaces and on gratings*. Springer, 1988.

- [2] HERMINGHAUS, S., KLOPFEISCH, M., AND SCHMIDT, H. Attenuated total reflectance as a quantum interference phenomenon. *Optics Letters* 19 (1994), 293–295.