CODES FOR CHAPTERS 2 AND 3

Chapters 2 and 3

fresnel.f, fresnel.cpp, fresnel.m
Subroutine `fresnel(n,k,th,rhos,rhop,rho)` calculates Fresnel reflectances from equation (2.113).
Input: `n` (= n) and `k` (= k) are real and imaginary parts of the complex index of refraction, and `th` (= θ) is the off-normal angle of incidence (in radians).
Output: `rhos` (= $\rho_\perp$) and `rhop` (= $\rho_\parallel$) are perpendicular and parallel-polarized reflectance, respectively, while `rho` (= ρ) is the unpolarized reflectance.

Chapter 3

emdiel.f90, emdiel.cpp, emdiel.m
Function `emdiel(n)` calculates the unpolarized, spectral, hemispherical emissivity of an optical surface of a dielectric material from equation (3.82).
Input: `n` (= n) refractive index of dielectric.

emmet.f90, emmet.cpp, emmet.m
Function `emmet(n,k)` calculates the unpolarized, spectral, hemispherical emissivity of an optical surface of a metallic material from equation (3.77).
Input: `n` (= n) and `k` (= k) are the real and imaginary parts of the metal's complex index of refraction.

callemdiel.f90, callemdiel.cpp, emmet.m, callemdiel.exe
Program `callemdiel` is a stand-alone front end for function `emdiel`, prompting for input (refractive index n) and returning the unpolarized, spectral, hemispherical as well as normal emissivities.

callemmet.f90, callemmet.cpp, callemmet.m, callemmet.exe
Program `callemmet` is a stand-alone front end for function `emmet`, prompting for input (complex index of refraction n, k) and returning the unpolarized, spectral, hemispherical as well as normal emissivities.

dirreflec.f, dirreflec.cpp, dirreflec.m, dirreflec.exe
Program `dirrecflec` is a stand-alone front end for subroutine `fresnel`, calculating reflectivities for various incidence angles. The user is prompted to input the complex index of refraction, n and k, and the (equal) spacing of incidence angles $\Delta\theta$ (in degrees); the program then returns perpendicular polarized, parallel polarized, and unpolarized reflectivities, as well as unpolarized emissivities.

totem.f90, totem.cpp, totem.m
Program `totem` is a routine to evaluate the total, directional or hemispherical emittance or absorptance of an opaque material, based on an array of spectral data, by 10-point Gaussian quadrature.
Input (by changing data in the heading of function `emlcl(y)`):

`N`	=	number of data points for spectral emittance,
`nrefr`	=	refractive index of adjoining material (`nrefr=1` for vacuum and gases),
`T`	=	temperature of material (for total emittance), or of gray irradiating source (for total absorptance), in K,

`lambda(N)` = `N` distinct wavelengths in ascending order, for which the spectral emittance is given, in μm,
`eps(N)` = `N` corresponding spectral emittances.
Output (printed to screen):
`emitt` = total directional or hemispherical emittance or absorptance.
Case 1: Total, directional emittance (`eps` contains spectral, directional values at temperature T):
From equation (3.8)

$$\epsilon'(T,\hat{\mathbf{s}}) = \frac{1}{n^2\sigma T^4}\int_0^\infty \epsilon'_\lambda(\lambda,T,\hat{\mathbf{s}})E_{b\lambda}(T)\,d\lambda = \int_0^1 \epsilon'_\lambda(\lambda(f),T,\hat{\mathbf{s}})\,df, \tag{CC-3-1}$$

where, from equation (1.23)

$$f(n\lambda T) = \int_0^\lambda \frac{E_{b\lambda}d\lambda}{n^2\sigma T^4}. \tag{CC-3-2}$$

In order to write equation (CC-3-1) in terms of blackbody fraction f, wavelength must be known as a function of f (for given n and T), i.e., equation (CC-3-2) must be inverted. The 10 values of $(n\lambda T)$, corresponding to the 10 Gaussian quadrature points $f_i(n\lambda T)$ have been precalculated (using function `bbfn`) and are stored in array `y(i)`. The total emittance is then calculated by expressing equation (CC-3-1) in quadrature form, or

$$\epsilon'(T,\hat{\mathbf{s}}) \simeq \sum_{i=1}^{10} \epsilon'_\lambda(\lambda_i,T,\hat{\mathbf{s}})w_i, \tag{CC-3-3}$$

where

$$\lambda_i = y_i/nT, \tag{CC-3-4}$$

and the w_i are Gaussian quadrature weights. This necessitates that ϵ'_λ must be known at very specific wavelengths, that are not ordinarily part of the given array. The "correct" value for ϵ'_λ is evaluated by linear interpolation between array values, assuming ϵ'_λ = const = `eps(1)` for λ_i <`lambda(1)`, and ϵ'_λ = const = `eps(N)` for λ_i >`lambda(N)`.
Case 2: Total, hemispherical emittance (`eps` contains spectral, hemispherical values at temperature T):
From equation (3.10)

$$\epsilon(T) = \frac{1}{n^2\sigma T^4}\int_0^\infty \epsilon_\lambda(\lambda,T)E_{b\lambda}\,d\lambda = \int_0^1 \epsilon_\lambda(\lambda(f),T)\,df \simeq \sum_{i=1}^{10}\epsilon_\lambda(\lambda_i,T)w_i. \tag{CC-3-5}$$

Thus, the calculation is identical to Case 1.
Case 3: Total, directional absorptance (`eps` contains spectral, directional values at the surface temperature T_s, irradiation is assumed to come from a gray source at temperature T).
From equations (3.23) and (3.31)

$$\alpha'(T_s,T,\hat{\mathbf{s}}) = \frac{1}{n^2\sigma T^4}\int_0^\infty \epsilon'_\lambda(\lambda,T,\hat{\mathbf{s}})E_{b\lambda}(T)\,d\lambda = \int_0^1 \epsilon'_\lambda(\lambda(f),T_s)\,df \simeq \sum_{i=1}^{10}\epsilon'_\lambda(\lambda_i,T_s)w_i, \tag{CC-3-6}$$

and the calculation is again identical.

Case 4: Total, hemispherical absorptance (`eps` contains spectral, hemispherical values at surface temperature T_s; irradiation is assumed to be gray and diffuse with source temperature T).
Then, from equations (3.27) and (3.31)

$$\begin{aligned}\alpha(T_s,T) &= \frac{1}{n^2\sigma T^4}\int_0^\infty \epsilon_\lambda(\lambda,T_s)E_{b\lambda}(T)\,d\lambda \\ &= \int_0^1 \epsilon_\lambda(\lambda(f),T_s)\,df \simeq \sum_{i=1}^{10}\epsilon_\lambda(\lambda_i,T_s)w_i. \end{aligned} \qquad \text{(CC-3-7)}$$

Examples
Two examples have been programmed into `totem` (or, rather, function `emlcl`):

1.: The material of Problem 3.1, with a step function in spectral emittance of

$$\epsilon_\lambda = \begin{cases} 0.5, & \lambda < 5\mu\text{m}, \\ 0.3, & \lambda > 5\mu\text{m}, \end{cases}$$

and a temperature of $T = 500$ K. For part *a*) `nrefr=1.0`, and for *b)* `nrefr=2.0` (implemented here) This results in `emitt=0.3435` for *a*) and `emitt=0.4296` for *b*).

2.: Aluminum oxide, as given in Fig. 1-14, discretized into eight equally-spaced values (commented out as given here). For temperature of $T = 500$ K and `nrefr=1.0` this results in `emitt=0.7494`.