CODES FOR CHAPTER 7

semigray.f90, semigray.cpp, semigray.m,
semigraydf.f90, semigraydf.cpp, semigraydf.m

Subroutine `semigray` provides the solution to equations (7.5) for an enclosure consisting of N diffusely emitting surfaces with diffuse and specular reflectance components, considering two spectral ranges (one for external irradiation, one for emission). For each surface the area, emittance and specular reflectance (two values each), external irradiation and either heat flux or temperature must be specified. In addition, the upper triangle of the view factor matrix must be provided for both spectral ranges (F^s_{i-j}; $i = 1, N$; $j = i, N$). For closed configurations, the diagonal view factors F^s_{i-i} are not required, since they can be calculated from the summation rule. The remaining view factors are calculated from reciprocity. On output, the program provides all view factors, and temperatures and radiative heat fluxes for all surfaces.

Input:

`N` = number of surfaces in enclosure

`iclsd` = closed or open configuration identifier
`iclsd`= 1: configuration is closed; diagonal F^s_{i-i} evaluated from summation rule
`iclsd`≠ 1: configuration has openings; F^s_{i-i} must be specified

`A(N)` = vector containing surface areas, [m^2]

`EPS(2,N)` = vector containing surface emittances for 2 spectral ranges

`RHOs(2,N)` = vector containing surface specular reflectance components for 2 spectral ranges

`HOs(N)` = vector containing external irradiation, in [W/m^2]

`Fs(2,N,N)` = vector containing view factors for 2 spectral ranges; on input only F^s_{i-j} with $j > i$ (`iclsd`=1) or $j \geq i$ (`iclsd`≠ 1) are required; remainder are calculated

`ID(N)` = vector containing surface identifier:
`ID`=0: surface heat flux is specified, in [W/m^2]
`ID`=1: surface temperature is specified, in [K]

`PIN(N)` = vector containing surface emissive powers (`id`=1) and fluxes (`id`=2)

Output:

`POUT(N)` = vector containing unknown surface fluxes (for surfaces with `id`=1) and emissive powers (for surfaces with `id`=0)

Subroutine `semigraydf` is a simplified version of subroutine `semigray` by assuming all surfaces to be diffuse, and input is changed by requiring `HO(N)` and `F(N,N)` (and no reflectance) instead of `RHOs(2,N)`, `HOs(N)` and `Fs(2,N,N)` (note that diffuse view factors do not depend on reflectance properties).

semigrxch.f90, semigrxch.cpp, semigrxch.m,
semigrxchdf.f90, semigrxchdf.cpp, semigrxchdf.m

Program `semigrxch` is a front end for subroutine `semigray` providing the necessary input for Example 7.1, primarily view factors calculated by calls to function `view`; similarly, program `semigrxchdf` is a front end for subroutine `semigraydf`. These programs may be used as a starting point for more involved radiative exchange problems.

bandapp.f90, bandapp.cpp, bandapp.m,
bandappdf.f90, bandappdf.cpp, bandapp.m

Subroutine bandapp provides the solution to equations (7.6) for an enclosure consisting of N diffusely emitting surfaces with diffuse and specular reflectance components, considering M spectral bands. For each surface the area, emittance, specular reflectance and external irradiation (one value for each spectral band), and either heat flux or temperature must be specified. In addition, the upper triangle of the view factor matrix must be provided for each spectral band (F^s_{i-j}; $i = 1, N$; $j = i, N$). For closed configurations, the diagonal view factors F^s_{i-i} are not required, since they can be calculated from the summation rule. The remaining view factors are calculated from reciprocity. On output, the program provides all view factors, and temperatures and radiative heat fluxes for all surfaces.

Input:

`M`	=	number of spectral bands
`N`	=	number of surfaces in enclosure
`iclsd`	=	closed or open configuration identifier `iclsd`= 1: configuration is closed; diagonal F^s_{i-i} evaluated from summation rule `iclsd`≠ 1: configuration has openings; F^s_{i-i} must be specified
`A(N)`	=	vector containing surface areas, [m^2]
`EPS(M,N)`	=	matrix containing surface emittances for M spectral ranges
`RHOs(M,N)`	=	matrix containing surface specular reflectance components for M spectral ranges
`HOs(M,N)`	=	matrix containing external irradiation for M spectral ranges, in [W/m^2]
`Fs(M,N,N)`	=	matrix containing view factors for M spectral ranges; on input only F^s_{i-j} with $j > i$ (`iclsd`=1) or $j \geq i$ (`iclsd`≠ 1) are required; remainder are calculated
`ID(N)`	=	vector containing surface identifier: `ID`=0: surface heat flux is specified, in [W/m^2] `ID`=1: surface temperature is specified, in [K]
`q(N)`	=	vector containing known surface fluxes (only for surfaces with `id`=2)
`T(N)`	=	vector containing known surface temperatures (only for surfaces with `id`=1)

Output:

`q(N)`	=	vector containing known surface fluxes (for all surfaces)
`T(N)`	=	vector containing known surface temperatures (for all surfaces)

Subroutine bandappdf is a simplified version of subroutine bandapp by assuming all surfaces to be diffuse, and input is changed by requiring `HO(M,N)` and `F(N,N)` (and no reflectance) instead of `RHOs(M,N)`, `HOs(M,N)` and `Fs(M,N,N)` (note that diffuse view factors do not depend on reflectance properties).

bandmxch.f90, bandmxch.cpp, bandmxch.m,
bandmxchdf.f90, bandmxchdf.cpp, bandmxch.m

Program bandmxch is a front end for subroutine bandapp providing the necessary input for Example 7.2, primarily view factors calculated by calls to function `view`; similarly, program bandmxchdf is a front end for subroutine bandappdf. These programs may be used as a starting point for more involved radiative exchange problems.