APPENDIX F

COMPUTER CODES

This appendix contains a listing and brief description of a number of computer programs that may be helpful to the reader of this book, and that can be downloaded from its dedicated website located at http://booksite.elsevier.com/9780123869449. Some of the codes are very basic and are entirely intended to aid the reader with the solution to the problems given at the end of the more basic chapters. Some of the codes were born out of research, but are basic enough to aid a graduate student with more complicated assignments or a semester project. And a few programs are so sophisticated in nature that they will be useful only to the practicing engineer conducting his or her own research. Finally, it is anticipated that the website will be kept up-to-date and augmented once in a while. Thus, there may be a few additional programs not described in this appendix.

It is a fact that most engineers have done, and still do, their programming in Fortran, and the author of this book is no exception. It is also true that computer scientists and most commercial programmers do their work in C++; more importantly, the younger generation of engineers at many universities across the U.S. are now also learning C++. Both compiled languages have in recent years been trumped by MATLAB® [1], which—while an interpreted rather than compiled language—has many convenient mathematical and graphical tools. Since all the programs in this listing were written by the author, either for research purposes or for the creation of this book, they all started their life in Fortran (older programs as Fortran77, and the later ones as Fortran90). However, as a gesture toward the C++ and MATLAB® communities, the most basic codes have all been converted to C++ as well as MATLAB®, as indicated below by the program suffixes `.cpp` and `.m`. If desired, all other programs are easily converted with freeware translators such as `f2c` (resulting in somewhat clumsy, but functional codes). Finally, self-contained programs that have been precompiled for Microsoft Windows have the suffix `.exe`.

The programs are listed in order by chapter in which they first appear. More detailed descriptions, sometimes with an example, can be found on the website. Third-party codes that are also provided at the website are listed at the end.

Chapter 1

`bbfn.f`, `bbfn.cpp`, `bbfn.m`:	Function `bbfn(x)` calculates the fractional blackbody emissive power, as defined by equation (1.23), where the argument is `x` $= n\lambda T$ with units of μm K.
`planck.f`, `planck.cpp`, `planck.m`, `planck.exe`:	`planck` is a small stand-alone program that prompts the user for input (temperature and wavelength or wavenumber), then calculates the spectral blackbody emissive powers $E_{b\lambda}/T^5$, $E_{b\eta}/T^3$ and the fractional blackbody emissive power $f(\lambda T)$.

Chapters 2 and 3

`fresnel.f`, `fresnel.cpp`, `fresnel.m`: Subroutine `fresnel` calculates Fresnel reflectivities from equation (2.113) for a given complex index of refraction and incidence angle.

Chapter 3

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

`emmet.f90`, `emmet.cpp`, `emmet.m`: Function `emmet` calculates the unpolarized, spectral, hemispherical emissivity of an optical surface of a metallic material from equation (3.77).

`callemdiel.f90`, `callemdiel.cpp`, `callemdiel.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 `dirreflec` is a stand-alone front end for subroutine `fresnel`, returning perpendicular polarized, parallel polarized, and unpolarized reflectances.

`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.

Chapter 4 and Appendix D

`view.f90`, `view.cpp`, `view.m`: A function to evaluate any of the 51 view factors given in Appendix D.

`parlplates.f90`, `parlplates.cpp` `parlplates.m`: A function to evaluate the view factor between two displaced parallel plates, as given by equation (4.42).

`perpplates.f90`, `perpplates.cpp`, `perpplates.m`: A function to evaluate the view factor between two displaced perpendicular plates, as given by equation (4.41).

`viewfactors.f90`, `viewfactors.cpp`, `viewfactors.m`, `viewfactors.exe`: A stand-alone front end to functions `view`, `parlplates`, and `perpplates`. The user is prompted to input configuration number and arguments; the program then returns the requested view factor.

Chapter 5

`graydiff.f90`, `graydiff.cpp`, `graydiff.m`: Subroutine `graydiff` provides the solution to equation (5.38) for an enclosure consisting of N gray-diffuse surfaces. For each surface the area, emittance, external irradiation, and either heat flux or temperature must be specified. In addition, the upper triangle of the view factor matrix must be provided (F_{i-j}; $i = 1, N$; $j = i, N$). For closed configurations, the diagonal view factors F_{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.

`graydiffxch.f90`, `graydiffxch.cpp`, `graydiffxch.m`: Program `graydiffxch` is a front end for subroutine `graydiff`, generating the necessary input parameters for a three-dimensional variation to Example 5.4, primarily view factors calculated by calls to function `view`. This program may be used as a starting point for more involved radiative exchange problems.

Chapter 6

`graydifspec.f90`, `graydifspec.cpp`, `graydifspec.m`: Subroutine `graydifspec` provides the solution to equation (6.23) for an enclosure consisting of N diffusely emitting surfaces with diffuse and specular reflectance components. For each surface the area, emittance, specular reflectance, external irradiation, and either heat flux or temperature must be specified. In addition, the upper triangle of the specular view factor matrix must be provided (F^s_{i-j}; $i = 1, N$; $j = i, N$). Otherwise same as `graydiff`.

`grspecxch.f90`, `grspecxch.cpp`, `grspecxch.m`: Program `grspecxch` is a front end for subroutine `graydifspec`, similar to `graydiffxch`.

Chapter 7

`semigray.f90`, `semigray.cpp`, `semigray.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. Otherwise same as `graydiff`.

`semigrxch.f90`, `semigrxch.cpp`, `semigrxch.m`: Program `semigrxch` is a front end for subroutine `semigray` providing the necessary input for Example 7.1. This program may be used as a starting point for more involved radiative exchange problems.

`bandapp.f90`, `bandapp.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. Otherwise same as `graydiff`.

`bandmxch.f90`, `bandmxch.cpp`, `bandmxch.m`: Program `bandmxch` is a front end for subroutine `bandapp` providing the necessary input for Example 7.2. This program may be used as a starting point for more involved radiative exchange problems.

Chapter 8

`MCintegral.f90`: `MCintegral` is a little program that evaluates the integral $\int_a^b f(x)\,dx$ for any specified function by the Monte Carlo method.

Chapter 11

`voigt.f`: Subroutine `voigt` calculates the spectral absorption coefficient for a Voigt-shaped line based on the fast algorithm by Humlíček [2], as a function of line intensity, and Lorentz and Doppler line widths.

`nbkdistdb.f90`: Program `nbkdistdb` is a Fortran90 code to calculate narrow band k-distributions for a number of temperatures and a number of wavenumber ranges, for a gas mixture containing CO_2, H_2O, CH_4, and soot. The spectral absorption coefficient is calculated directly from the HITRAN or HITEMP databases.

`nbkdistsg.f90`: Program `nbkdistsg` is a Fortran90 code to calculate a single narrow band k-distribution from a given array of wavenumber–absorption coefficient pairs.

`wbmxxx.f`, `wbmxxxcl.f`, `wbmxxxcl.exe`: Subroutines `wbmxxx`, where `xxx` stands for the different gases `h20`, `co2`, `ch4`, `co`, `no`, and `so2`, calculate for a given temperature the ratios $\Psi^*(T)/\Psi^*(T_0)$ [from equations (11.144) and (11.148)] and $\Phi(T)/\Phi(T_0)$ [from equation (11.149)], i.e., the functions shown in Figs. 11-23 through 11-25. The stand-alone programs `wbmxxxcl.f` are front ends for the `wbmxxx.f`, prompting the user for input, and printing the ratios $\Psi^*(T)/\Psi^*(T_0)$ and $\Phi(T)/\Phi(T_0)$ to the screen for all bands listed in Table 11.3.

`emwbm.f`, `ftwbm.f`, `wangwbm.f`: Fortran functions to calculate the nondimensional total band absorptance A^* from the Edwards and Menard model, Table 11.2 (`emwbm`), the Felske and Tien model, equation (11.156) (`ftwbm`), and the Wang model, equation (11.158) (`wangwbm`).

`wbmodels.f`, `wbmodels.exe`: Stand-alone front end for the functions `emwbm`, `ftwbm`, and `wangwbm`; the nondimensional total band absorptance A^* is printed to the screen, as calculated from three band models (Edwards and Menard, Felske and Tien, and Wang models).

`wbmkvsg.f`: Fortran subroutine `wbmkvsg` calculates the κ^* *vs.* g^* distribution of equation (11.170).

`totemiss.f`: Fortran subroutine `totemiss` calculates the total emissivity of an isothermal gas mixture, using Leckner's model, equations (11.177) through (11.181).

`totabsor.f`: Fortran subroutine `totabsor` calculates the total absorptivity of an isothermal gas mixture, using Leckner's model, equations (11.177) through (11.181).

`Leckner.f`, `Leckner.exe`: Stand-alone front end for `totemiss` and `totabsor`, with total emissivities and absorptivities printed to the screen.

Chapter 12

`coalash.f90`: This file contains subroutine `coalash` (plus a front end for screen input and output) to determine nondimensionalized spectral absorption and extinction coefficients κ^* and β^*, as listed in Table 12.3, from the Buckius and Hwang [3] and the Mengüç and Viskanta [4] models, as functions of complex index of refraction $m = n - ik$ and size parameter x.

`mmmie.f`: Program `mmmie` calculates Mie coefficients (scattering coefficients a_n and b_n, efficiencies Q_{sca}, Q_{ext}, and Q_{abs}, and asymmetry factor g; see Section 12.2 for definitions), and relates them to particle cloud properties (extinction coefficient β, absorption coefficient κ, scattering coefficient σ_s, cloud asymmetry factor g, scattering phase function Φ for specified scattering angles, and phase function expansion coefficients A_n, as defined in Section 12.3).

Chapter 16

`P1sor.f90`, `P1sor.cpp`: Subroutine `P1sor` provides the solution to equation (16.38) with its boundary condition (16.49) for a two-dimensional (rectangular or axisymmetric cylinder) enclosure with reflecting walls and an absorbing, emitting, linear-anisotropically scattering medium. For each surface the emittance and blackbody intensities must be specified; for the medium spatial distributions of radiation properties and blackbody intensities must be input. Internal incident radiation (G) and wall flux (q) fields are calculated. Can be used for gray problems or on a spectral basis.

`P1-2D.f90`, `P1-2D.cpp`: Program `P1-2D` is a front end for subroutine `P1sor`, setting up the problem for a gray medium with spatially constant radiative properties; it may be used as a starting point for more involved applications.

`Delta.f90`: Program `Delta` is a stand-alone program to calculate the rotation matrix $\Delta^n_{mm'}(\alpha, \beta, \gamma)$ required for the boundary conditions of higher-order P_N-approximations, as given by equations (16.64) through (16.67).

`pnbcs.f90`: Program `pnbcs` is a stand-alone program to calculate the Legendre half-moments $p^m_{n,j}$ and coefficients $u^m_{li}, v^m_{li}, w^m_{li}$, which are required for the boundary conditions of higher-order P_N-approximations, as given by equations (16.71) through (16.72).

Chapter 19

`transPN.f90`: Program `transPN` calculates energy from a pulsed collimated laser source transmitted through an absorbing, isotropically scattering slab as a function of time, using the P_1 and $P_{1/3}$ methods.

Chapter 20

`fskdist.f90`: Program `fskdist` is a Fortran90 code to calculate full-spectrum k-distributions for a number of Planck function temperatures and a single gas property state (temperature, partial and total pressures), for a gas mixture containing CO_2, H_2O, CH_4, and soot; weight functions $a(T, T_0, g)$ are calculated, as well. The spectral absorption coefficient is either calculated directly from the HITRAN or HITEMP databases, or is supplied by the user.

`fskdco2.f90`, `fskdh2o.f90`: These subroutines determine full spectrum cumulative k-distributions for CO_2 and H_2O, respectively, employing the correlations of Modest and Mehta [5] and of Modest and Singh [6].

`fskdco2dw.f90`, `fskdh2odw.f90`: Equivalent to `fskdco2.f90` and `fskdh2o.f90`, but employing the older correlations of Denison and Webb [7, 8].

`kdistmix.f90`: Subroutine `kdistmix` finds the cumulative k-distribution for an n-component mixture from a given set of individual species cumulative k-distributions (narrow band, wide band, or full spectrum), employing the mixing scheme of Modest and Riazzi [9].

`fskdistmix.f90`: This Fortran90 routine finds the full spectrum cumulative k-distribution for a CO_2–H_2O mixture, employing the correlations of Modest and Mehta [5] and Modest and Singh [6], using one of three mixing schemes described by equations (20.162) (superposition), (20.163) (multiplication), or (20.167) (uncorrelated mixture).

Chapter 21

`mocacyl.f`, `rnarray.f`: Program `mocacyl` is a Monte Carlo routine for a nongray, nonisothermal, isotropically scattering medium confined inside a two-dimensional, axisymmetric cylindrical enclosure bounded by nongray, diffusely emitting and reflecting walls. Temperature and radiative properties are assumed known everywhere inside the enclosure and along the walls. Requires use of program `rnarray` to set up random number relationships (locations and wavenumbers of emission *vs.* random numbers). Calculates internal radiative heat sources $\nabla \cdot q^R$ as well as local radiative fluxes to the walls q_w^R.

`FwdMCcs.f90`, `FwdMCck1.f90`, `FwdMCck2.f90`: Program `FwdMCcs` is a standard forward Monte Carlo code for a narrow collimated beam penetrating through a nonabsorbing, isotropically scattering slab, calculating the flux onto a small, directionally selective detector, as given in Example 21.3. `FwdMCck1` and `FwdMCck2` are forward Monte Carlo codes for the same problem, but also allow for absorption in the medium; `FwdMCck1` uses standard ray tracing, while `FwdMCck2` uses energy partitioning; see Example 21.4.

`FwdMCps.f90`: Program `FwdMCps` is a standard forward Monte Carlo code for a radiative energy emitted by a point source penetrating through a nonabsorbing, isotropically scattering slab, calculating the flux onto a small, directionally selective detector.

`RevMCcs.f90`, `RevMCck1.f90`, `RevMCck2.f90`: These programs are backward Monte Carlo implementations of the equivalent `FwdMCcs`, `FwdMCcka1`, and `FwdMCcka2`, as also discussed in Examples 21.3 and 21.4.

`RevMCps.f90`: The backward Monte Carlo equivalent of `FwdMCps`.

Software Packages

`MONT3D` This code, developed at Colorado State University by Burns et al. [10–14], calculates radiative exchange factors for complicated, three-dimensional geometries by the Monte Carlo method, as given by equations (8.15) and (8.21). Diffuse and specular view factors may be calculated as special cases.

`VIEW3D` This code, developed at National Institute of Standards and Technology (NIST) by Walton [15], calculates radiative view factors with obstructions by adaptive integration.

`RADCAL` This code, developed at NIST by Grosshandler [16, 17], is a narrow band database for combustion gas properties, using tabulated values and theoretical approximations.

`EM2C` This package contains a number of Fortran codes, developed at the Ecole Centrale de Paris by Soufiani and Taine [18], calculating statistical narrow band properties as well as narrow band k-distributions for CO_2 and H_2O, using the HITRAN92 database together with some proprietary French high-temperature extensions.

`NBKDIR` This package contains a number of Fortran codes, developed at the Pennsylvania State University and the University of California at Merced by the author and his students/postdocs A. Wang, G. Pal, and J. Cai, for the assembly of full spectrum k-distributions from a narrow band k-distributions database. At the time of printing `NBKDIR` contained data for five species ($CO_2, H_2O, CO, CH_4, C_2H_4$), as well as nongray soot, for temperatures up to 3000 K and pressures up to 80 bar. Spectroscopic data are taken from the HITEMP 2010 (CO_2, H_2O, CO) [19] and HITRAN 2008 (CH_4, C_2H_4) [20].

`FVM2D` This Fortran77 code, developed at the University of Minnesota and Nanyang Technological University by Chai and colleagues [21–23], calculates radiative transfer in participating media using the finite-volume method of Chapter 17 for a two-dimensional, rectangular enclosure with reflecting walls and an absorbing, emitting, anisotropically scattering medium. For each surface the emittance and blackbody intensities must be specified; for the medium spatial distributions of radiation properties and blackbody intensities must be input. Internal incident radiation (G) and wall flux (q) fields are calculated. Can be used for gray problems or on a spectral basis.

References

1. *MathWorks* MATLAB website, http://www.mathworks.com/products/matlab/.
2. Humlíček, J.: "Optimized computation of the Voigt and complex probability functions," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 27, p. 437, 1982.
3. Buckius, R. O., and D. C. Hwang: "Radiation properties for polydispersions: Application to coal," *ASME Journal of Heat Transfer*, vol. 102, pp. 99–103, 1980.
4. Mengüç, M. P., and R. Viskanta: "On the radiative properties of polydispersions: A simplified approach," *Combustion Science and Technology*, vol. 44, pp. 143–159, 1985.
5. Modest, M. F., and R. S. Mehta: "Full spectrum k-distribution correlations for CO_2 from the CDSD-1000 spectroscopic databank," *International Journal of Heat and Mass Transfer*, vol. 47, pp. 2487–2491, 2004.
6. Modest, M. F., and V. Singh: "Engineering correlations for full spectrum k-distribution of H_2O from the HITEMP spectroscopic databank," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 93, pp. 263–271, 2005.
7. Denison, M. K., and B. W. Webb: "Development and application of an absorption line blackbody distribution function for CO_2," *International Journal of Heat and Mass Transfer*, vol. 38, pp. 1813–1821, 1995.
8. Denison, M. K., and B. W. Webb: "An absorption-line blackbody distribution function for efficient calculation of total gas radiative transfer," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 50, pp. 499–510, 1993.
9. Modest, M. F., and R. J. Riazzi: "Assembly of full-spectrum k-distributions from a narrow-band database; effects of mixing gases, gases and nongray absorbing particles, and mixtures with nongray scatterers in nongray enclosures," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 90, no. 2, pp. 169–189, 2005.
10. Burns, P. J.: "MONTE–a two-dimensional radiative exchange factor code," Technical report, Colorado State University, Fort Collins, 1983.
11. Maltby, J. D.: "Three-dimensional simulation of radiative heat transfer by the Monte Carlo method," M.S. thesis, Colorado State University, Fort Collins, CO, 1987.
12. Burns, P. J., and J. D. Maltby: "Large-scale surface to surface transport for photons and electrons via Monte Carlo," *Computing Systems in Engineering*, vol. 1, no. 1, pp. 75–99, 1990.
13. Maltby, J. D., and P. J. Burns: "Performance, accuracy and convergence in a three-dimensional Monte Carlo radiative heat transfer simulation," *Numerical Heat Transfer – Part B: Fundamentals*, vol. 16, pp. 191–209, 1991.
14. Zeeb, C. N., P. J. Burns, K. Branner, and J. S. Dolaghan: "User's manual for Mont3d – Version 2.4," Colorado State University, Fort Collins, CO, 1999.
15. Walton, G. N.: "Calculation of obstructed view factors by adaptive integration," Technical Report NISTIR–6925, National Institute of Standards and Technology (NIST), Gaithersburg, MD, 2002.
16. Grosshandler, W. L.: "Radiative transfer in nonhomogeneous gases: A simplified approach," *International Journal of Heat and Mass Transfer*, vol. 23, pp. 1447–1457, 1980.

17. Grosshandler, W. L.: "RADCAL: a narrow-band model for radiation calculations in a combustion environment," Technical Report NIST Technical Note 1402, National Institute of Standards and Technology, 1993.
18. Soufiani, A., and J. Taine: "High temperature gas radiative property parameters of statistical narrow-band model for H_2O, CO_2 and CO, and correlated-*k* model for H_2O and CO_2," *International Journal of Heat and Mass Transfer*, vol. 40, no. 4, pp. 987–991, 1997.
19. Rothman, L. S., I. E. Gordon, R. J. Barber, H. Dothe, R. R. Gamache, A. Goldman, V. I. Perevalov, S. A. Tashkun, and J. Tennyson: "HITEMP, the high-temperature molecular spectroscopic database," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 111, no. 15, pp. 2139–2150, 2010.
20. Rothman, L. S., I. E. Gordon, A. Barbe, D. C. Benner, P. F. Bernath, M. Birk, V. Boudon, L. R. Brown, A. Campargue, J.-P. Champion, K. Chance, L. H. Coudert, V. Dana, V. M. Devi, S. Fally, J.-M. Flaud, R. R. Gamache, A. Goldman, D. Jacquemart, I. Kleiner, N. Lacome, W. J. Lafferty, J.-Y. Mandin, S. T. Massie, S. N. Mikhailenko, C. E. Miller, N. Moazzen-Ahmadi, O. V. Naumenko, A. V. Nikitin, J. Orphal, V. I. Perevalov, A. Perrin, A. Predoi-Cross, C. P. Rinsland, M. Rotger, M. Simeckova, M. A. H. Smith, K. Sung, S. A. Tashkun, J. Tennyson, R. A. Toth, A. C. Vandaele, and J. V. Auwera: "The HITRAN 2008 molecular spectroscopic database," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 110, pp. 533–572, 2009.
21. Chai, J. C., H. S. Lee, and S. V. Patankar: "Finite volume method for radiation heat transfer," *Journal of Thermophysics and Heat Transfer*, vol. 8, no. 3, pp. 419–425, 1994.
22. Chai, J. C., H. S. Lee, and S. V. Patankar: "Treatment of irregular geometries using a Cartesian coordinates finite-volume radiation heat transfer procedure," *Numerical Heat Transfer – Part B: Fundamentals*, vol. 26, pp. 225–235, 1994.
23. Chai, J. C., G. Parthasarathy, H. S. Lee, and S. V. Patankar: "Finite volume method radiative heat transfer procedure for irregular geometries," *Journal of Thermophysics and Heat Transfer*, vol. 9, no. 3, pp. 410–415, 1995.