Supplement 1

IPR 1.0: an efficient method for calculating solar radiation absorbed by individual plants in sparse heterogeneous woody plant communities

The C++ Code

(Version 1.0, November 2013)

Yu Zhang, Wenjun Chen, and Junhua Li

Canada Centre for Mapping and Earth Observation, Natural Resources Canada, Ottawa, Canada
Correspondence to: Y. Zhang (yu.zhang@nrcan.gc.ca)

The code was developed and tested using Microsoft Visual C++ (Win32 Console Applications). In addition to the code, an example input dataset and the model output were included as well. Users can copy and save the included input dataset to a text file (using the name input.txt) to test the code. For simplicity, the input and output files are directly put under the code folder. More detailed description of the input and output files can be found in the User's Manual (Supplement 2). The description of the method and equations can be found in the accompanied paper.

The code was developed to calculate the solar radiation absorbed by individual plants of a plant community at different times in a day. The input data include the latitude of the location, the day of the year, the conditions of the herb and woody plant strata, and the solar radiation (direct and diffuse) on a horizontal surface above the plant canopy at different times in the day. The outputs include the fractions of sunlit leaf area of the woody plant strata and the herb stratum, the solar radiation of the sunlit and shaded leaves. The model also calculates the fraction of sunlit area of the ground, and the solar radiation in the sunlit and shaded areas of the ground, and the average solar radiation of the ground. Users can use these outputs to calculate photosynthesis, transpiration and energy balance of the individual plants and the energy balance of the ground.

-------------------------------Beginning of the code---

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

#define Pi 3.1415926
#define MaxPlants 5000
#define Max_XX    100   //maximum length considered (m)

#define BB 18      //steps indeclination angle from 0 to Pi/2
#define TTmax 48   //maximum number of solar radiations input data in a day
#define NNmax 5    //maximum number of woody strata

void Input_Check(int NN, double *Density, double *HH, double *hh, double *DD);
void Fraction_Overlap(int NN, double *HH, double *hh, double *Frac_Overlap[]);

void Daily_IPR(FILE *fpt_out, int TT, int NN, double *B0, double dB, double *Time, double *B,
         double *I_b, double *I_b0, double *I_d0, double LAI_h, double alfa_g, double alfa_h,
         double *alfa,  double Kh, double *K, double *Density, double *HH,
         double *hh, double *DD, double *LAI);

double Calculate_Sunlit_Frac(int NN, double B, int I0, double *Foverlap[], double *K,
         double *Density, double *HH, double *hh, double *DD, double *Leaf_area,
         double *LA_density,double *F1a_ijB, double *fsa_i_B);

double Shading_effects(double Z, double H1, double H2, double DD, double SL,
         double Kp, double Distance, double pj, double sinB, double cosB, double tanB);

double DayLength_SunElevation(int TT, double Lat, double DOY, double *Time,
         double *Sun_Altitude, double *Sun_RiseSet);
double Interpolate(double *XX, double *YY, double X);

int main()
{
//The IPR model: An Individual Plant Radiation model
//(Version 1.0, November 2013)
```

```
//Developed by Yu Zhang, Wenjun Chen and Junhua Li
//Canada Centre for Mapping and Earth Observation, Natural Resources Canada

  char A[200], Path[]= ""; //the path for the model input/output. Users can modifiy it.
  FILE *fpt_in, *fpt_out;

  int i, T, TT, N;
  int NN, DOY;
  double Lat, LAI_h,  alfa_g, alfa_h, Omega_h;
  double alfa[NNmax], Omega[NNmax], HH[NNmax], hh[NNmax], DD[NNmax], LAI[NNmax], Density[NNmax];

  double Kh, K[NNmax];
  double B0[TTmax], dB;
  double Day_Length, Sun_RiseSet[2], Time[TTmax], B[TTmax], Frac_Beam[TTmax],
         I_b[TTmax], I_b0[TTmax], I_d0[TTmax];
  double Total_I, F_Diffuse;

  sprintf(A,"%sOutput.txt", Path);  //the name of the output file. Users can modify it
  if((fpt_out = fopen(A, "w")) == NULL) {
      printf("Cannot create output file %s\n", A);
      exit(0);
  }

  sprintf(A,"%sInput.txt", Path); //the name of the input file. Users can modify it
  if((fpt_in = fopen(A, "r")) == NULL) {
      printf("Cannot open input file %s\n", A);
      exit(0);
  }
 fgets(A, 199, fpt_in);  //a note line for the input file

 fscanf(fpt_in, "%lf", &Lat); fgets(A, 199, fpt_in);  //Latitude in degrees
 if(Lat>90 || Lat < -90) {
            printf("The Latitude of the site (%lf) should be between -90 to 90 degrees\n", Lat);
            exit(1);
  }

 fscanf(fpt_in, "%lf", &LAI_h);       fgets(A, 199, fpt_in);  //LAI of the herb stratum
 if(LAI_h < 0) {
```

```
            printf("The LAI of herb (%lf) should be >0\n", LAI_h);
            exit(1);
  }
 if(LAI_h<1.0e-10) {
        fgets(A, 199, fpt_in);  //line 4
        fgets(A, 199, fpt_in);  //line 5
        alfa_h = 0.001;
        Omega_h = 1.0;
 }
 else {
       fscanf(fpt_in, "%lf", &alfa_h); fgets(A, 199, fpt_in); //The absorption coefficient of herb stratum
       if(alfa_h < 0 || alfa_h>1.0) {
                  printf("The absorption coefficient of herb (%lf) should be between 0 and 1\n", alfa_h);
                  exit(1);
        }

       fscanf(fpt_in, "%lf", &Omega_h); fgets(A, 199, fpt_in);  //The clumping index of the herb stratum
       if(Omega_h < 0) {
                  printf("The clumping index of herb (%lf) should be >0\n", Omega_h);
                  exit(1);
        }
}

 fscanf(fpt_in, "%lf", &alfa_g);      fgets(A, 199, fpt_in);  //The absorption coefficient of the ground
 if(alfa_g < 0 || alfa_g>1.0) {
            printf("The absorption coefficient of the ground (%lf) should be between 0 and 1\n", alfa_g);
            exit(1);
  }

  fscanf(fpt_in, "%d", &NN);  fgets(A, 199, fpt_in);  //The number of woody strata
  if(NN>NNmax || NN < 0) {
            printf("The number of woody strata (%d) is out of the range (0 - %d) defined in the code\n",
                    NN, NNmax);
            exit(1);
  }

  if(NN == 0) {
            for(i = 0; i<7; i++) fgets(A, 199, fpt_in);  //read seven lines (lines 8 - 14)
```

```
 }
 else {
       for(i = 0; i<NN; i++) fscanf(fpt_in, "%lf", &HH[i]);  //top height, m
       fgets(A, 199, fpt_in);

       for(i = 0; i<NN; i++) fscanf(fpt_in, "%lf", &hh[i]);  //Bottom height, m
       fgets(A, 199, fpt_in);

       for(i = 0; i<NN; i++) fscanf(fpt_in, "%lf", &DD[i]);  //crown width, m
       fgets(A, 199, fpt_in);

       for(i = 0; i<NN; i++) fscanf(fpt_in, "%lf", &LAI[i]);  //local LAI
       fgets(A, 199, fpt_in);

       for(i = 0; i<NN; i++) fscanf(fpt_in, "%lf", &Density[i]);  //Plant density, plants/m2
       fgets(A, 199, fpt_in);

       for(i = 0; i<NN; i++) fscanf(fpt_in, "%lf", &alfa[i]);  //absorption coefficent
       fgets(A, 199, fpt_in);

       for(i = 0; i<NN; i++) fscanf(fpt_in, "%lf", &Omega[i]);  //clumping index
       fgets(A, 199, fpt_in);

       for(i = 0; i<NN; i++) {
             if(HH[i] < 0) {
                  printf("Stratum %d: The top height (%lf) should be >0 m \n", i+1, HH[i]);
                  exit(1);
             }
             if(hh[i] < 0) {
                  printf("Stratum %d: The bottom height (%lf) should be >0 m \n", i+1, hh[i]);
                  exit(1);
             }
             if(hh[i]>=HH[i]) {
printf("Stratum %d: The bottom height of crown (%lf) should be lower than the top of the crown (%lf)\n",
         i+1, hh[i], HH[i]);
                 exit(1);
             }
             if(DD[i] < 0) {
```

```
                  printf("Stratum %d: The crown width(%lf) should be >0 m \n", i+1, DD[i]);
                  exit(1);
             }
             if(LAI[i] < 0) {
                  printf("Stratum %d: The local LAI(%lf) should be >0 m \n", i+1, LAI[i]);
                  exit(1);
             }
             if(alfa[i] < 0 || alfa[i]>1.0) {
                   printf("Stratum %d: The absorption coefficient (%lf) should be between 0 and 1\n",
                           i+1, alfa[i]);
                   exit(1);
             }
             if(Omega[i] < 0) {
                   printf("Stratum %d: The clumping index (%lf) should be >0\n", i+1, Omega[i]);
                   exit(1);
             }
       }
 }  //end of else

 fscanf(fpt_in, "%d", &DOY); fgets(A, 199, fpt_in);  //day of a year
 if(DOY>366 || DOY < 1) {
           printf("The day of year (%d) should be between 1 to 366\n", DOY);
           exit(1);
 }

fscanf(fpt_in, "%d", &TT);     fgets(A, 199, fpt_in);  //Number of solar radiation observations in the day
if(TT < 1 || TT >= TTmax) {
           printf("The number of radiation data (%d) is out of the range defined in the code (1 - %d)\n",
                   TT, TTmax);
           exit(1);
}
fgets(A, 199, fpt_in);   //a note line
for(T=0; T<TT; T++) {
      fscanf(fpt_in, "%lf%lf%lf", &Time[T], &Total_I, &F_Diffuse);
      if(Time[T]<0 || Time[T] >24) {
            printf("Radiation data line %d: The range of time (%lf) should be 0-24\n", T+1, Time[T]);
            exit(1);
      }
```

```
       if(Total_I<0) {
             printf("Radiation data line %d: The total radiation (%lf) should be >0\n", T+1, Total_I);
             exit(1);
       }
       if(F_Diffuse<0 || F_Diffuse>1) {
             printf("Radiation data line %d: The fraction of diffuse radiation (%lf) should be 0 - 1\n",
                     T+1, F_Diffuse);
             exit(1);
       }

       Frac_Beam[T] = 1 - F_Diffuse;
       I_b0[T] = Total_I * (1-F_Diffuse);
       I_d0[T] = Total_I * F_Diffuse;
 }

  dB = 90.0/BB * Pi/180.0;;
  B0[0] = 0.5*dB;
  for(T=1; T<BB; T++) B0[T] = B0[T-1] + dB;

  Kh = 0.5*Omega_h;
  for(N=0; N<NN; N++) K[N] = 0.5*Omega[N];
  Input_Check(NN, Density, HH, hh, DD);

  Day_Length=DayLength_SunElevation(TT, Lat, DOY, Time, B, Sun_RiseSet);

  for(T=0; T<TT; T++) {  //TT
        if(B[T]>0.1) {
              I_b[T] = I_b0[T]/sin(B[T]); //beam, in normal direction
        }
        else I_b[T] = 0.0;
  }

  Daily_IPR(fpt_out, TT, NN, B0, dB, Time, B, I_b, I_b0, I_d0, LAI_h, alfa_g, alfa_h,
                alfa, Kh, K, Density, HH, hh, DD, LAI);

  fclose(fpt_out);
return 1;
}
```

```
//=====================================================
void Daily_IPR(FILE *fpt_out, int TT, int NN, double *B0, double dB, double *Time, double *B,
           double *I_b, double *I_b0, double *I_d0, double LAI_h, double  alfa_g, double alfa_h,
           double *alfa, double Kh, double *K, double *Density, double *HH,
               double *hh,   double *DD, double *LAI)
{
  int N, i, j, T;

  double Leaf_area[NNmax], LA_density[NNmax], Fcrown_area[NNmax];
  double Foverlap[NNmax][NNmax], *pFoverlap[NNmax];

  double I_b_i[NNmax], I_b_h;

  double Fsunlit0[NNmax][BB], Fsunlit[NNmax], Fd_i[NNmax],
           F1a_ij[NNmax][NNmax], F1a_ijB[NNmax],
             fs_i[NNmax], fsa_i_B[NNmax], Exp_LA[NNmax],
             PLai[NNmax], r_i[NNmax];

  double I_scatt0[NNmax], I_scatt_inter[NNmax], I_scatt_intra[NNmax],
         I_sunlit[NNmax], I_shaded[NNmax];

  double l_ai, F2w_0, I_s2a_i, I_s2b_i[NNmax], I_s2c_i, F2w[BB];

  double Fsunlit_h0[BB], Fsunlit_h, Fd_h, Fd0_h, r_h,
           I_scatt0_h, I_scatt_h, Is1_h, I_sunlit_h, I_shaded_h;

  double Fsunlit_g0[BB], Fsunlit_g, Fd_g, Fd0_g,
           I_scatt_g, I_sunlit_g, I_shaded_g,  I_avg_g;

  double sinB, cosB, tanB, F, F1;

//--print head of the output
  fprintf(fpt_out, "Time    B   I_Total I_Direct I_Diffuse " );
  for(N=0; N<NN; N++) {
       fprintf(fpt_out, "Woody_Stratum_%d            ", N+1);
  }
  fprintf(fpt_out, "Herb_Stratum                ");
  fprintf(fpt_out, "Ground \n");
```

```
  fprintf(fpt_out, "----   ---  ------- -------- --------  ");
  for(N=0; N<NN; N++) {
       fprintf(fpt_out, "F_Sunlit I_Sunlit I_Shaded ");
  }
  fprintf(fpt_out, "F_Sunlit I_Sunlit I_Shaded  ");
  fprintf(fpt_out, "F_Sunlit I_Sunlit I_Shaded I_Ground_Avg\n");

//============Sunlit fractions for each elevation angle from 0 to 90
  for(N=0; N<NN; N++) {
    Leaf_area[N] = LAI[N]*DD[N]*DD[N];   //Leaf Area
      LA_density[N] = Leaf_area[N]/(DD[N]*DD[N]*(HH[N]-hh[N]));
  }

  for(N=0; N<NN; N++) pFoverlap[N] = &Foverlap[N][0];
  Fraction_Overlap(NN, HH, hh, pFoverlap);

  for(N=0; N<NN; N++) {
      Fd_i[N] = 0;
      fs_i[N] = 0;
      for(i=0; i<NN; i++) F1a_ij[N][i] = 0;
  }
  Fd_h = 0;
  Fd_g = 0;
  Fd0_h = 0;
  Fd0_g = 0;

  for(T=0; T<BB; T++) {
      sinB = sin(B0[T]);
      cosB = cos(B0[T]);
      tanB = sinB/cosB;

//woody strata
      F2w_0 = 0;
      for(N=0; N<NN; N++) {
        Fsunlit0[N][T] = Calculate_Sunlit_Frac(NN, B0[T], N, pFoverlap, K,
                           Density, HH, hh, DD,  Leaf_area, LA_density, F1a_ijB, fsa_i_B);
```

```
        Fd_i[N] += 2* K[N] * Fsunlit0[N][T] * cosB * dB;
//areal fraction of shade casted on the ground
        F2w_0 += Fsunlit0[N][T] * Leaf_area[N] * K[N] * Density[N]/sinB;

        fs_i[N]   += fsa_i_B[N] * dB;
      for(j=0; j<NN; j++) F1a_ij[N][j] += (1-F1a_ijB[j]) * dB;
      }  //loop N
    if(F2w_0 > 1) F2w_0 = 1;
      F2w[T] = 1 - F2w_0;

//--herb stratum
      F = sinB/(Kh*(LAI_h+0.0001))*(1-exp(-Kh * LAI_h/sinB));
    Fsunlit_h0[T] = F2w[T] * F;
      Fd0_h += 2 * Kh * F * cosB * dB;
      Fd_h  += 2*Kh*Fsunlit_h0[T]*cosB * dB;

//--on the ground
      F = exp(-Kh*LAI_h/sinB);
    Fsunlit_g0[T] = F2w[T]*F;
      Fd0_g += 2 * F * sinB * cosB * dB;
      Fd_g  += 2 * Fsunlit_g0[T] * sinB * cosB * dB;
  }  //loop T

  for(N=0; N<NN; N++) {
      Fcrown_area[N] = Density[N] * DD[N]* DD[N];

      l_ai = (HH[N]-hh[N])/6 + DD[N]/3;
      Exp_LA[N] = exp(-K[N] * LA_density[N] * l_ai);
      PLai[N] = LA_density[N] * l_ai;
      r_i[N] = 1 - exp(-K[N]*PLai[N]);
  }
  r_h = 0.88*(1-exp(-0.7*pow(LAI_h,0.75)));

//============== radiation
  for(T=0; T<TT; T++) {
      if(I_b0[T]+I_d0[T]>0) {

         for(N=0; N<NN; N++) I_b_i[N] = I_b[T] * K[N]; //beam light intercepted
```

```
         I_b_h = I_b[T] * Kh;
//--woody strata
         for(N=0; N<NN; N++) {
              Fsunlit[N] = Interpolate(B0, Fsunlit0[N], B[T]);
//scatter light generated in a crown (per unit leaf area)
              I_scatt0[N]=(I_b_i[N]*Fsunlit[N] + I_d0[T]*Fd_i[N])*(1-alfa[N]);
              I_s2a_i = I_scatt0[N]*(1-r_i[N])/(1-(1-alfa[N])*r_i[N]);
              I_s2b_i[N] = Leaf_area[N]*I_s2a_i/(4*DD[N]*(HH[N]-hh[N]) + 2*DD[N]*DD[N]);
            }  //loop N

            for(N=0; N<NN; N++) {
              I_scatt_intra[N] = I_scatt0[N]*(1-r_i[N])/(1-(1-alfa[N])*r_i[N]);

              F=0;
              F1 = 0;
              for(i=0; i<NN; i++) {
                F += (1 - F1a_ij[N][i])*I_s2b_i[i];
                  F1 += 1 - F1a_ij[N][i];
              }  //loop i
              I_s2c_i = F/(F1+0.0001);
              I_scatt_inter[N] = 0.5 * I_s2c_i * fs_i[N]/(2*PLai[N]);//0.5:only above hamisphere scatters

              I_sunlit[N]=(I_b_i[N]+I_d0[T]*Fd_i[N])*alfa[N]+I_scatt_inter[N]+I_scatt_intra[N];
              I_shaded[N]=          I_d0[T]*Fd_i[N] *alfa[N]+I_scatt_inter[N]+I_scatt_intra[N];
            }  // loop N
//--herb stratum
        Fsunlit_h = Interpolate(B0, Fsunlit_h0, B[T]);
//          scatter light generated in per unit leaf area)
            I_scatt0_h = (I_b_h*Fsunlit_h + I_d0[T]*Fd_h) * (1-alfa_h);

            Is1_h = 0;
            for(N=0; N<NN; N++) {
              Is1_h += (0.5 * (I_scatt0[N]-I_scatt_intra[N]) - I_scatt_inter[N]) * Leaf_area[N]*Density[N];
            }

            I_scatt_h = alfa_h * Is1_h * Fd0_h + I_scatt0_h * (1-alfa_h)*r_h/(1-(1-alfa_h)*r_h);

            I_sunlit_h = (I_b_h*Fsunlit_h + I_d0[T]*Fd_h)*alfa_h + I_scatt_h;
```

```
            I_shaded_h =                        I_d0[T]*Fd_h *alfa_h + I_scatt_h;

//--ground
            Fsunlit_g = Interpolate(B0, Fsunlit_g0, B[T]);

            I_scatt_g = Is1_h * Fd0_g + 0.5*I_scatt0_h*(1-r_h)/(1-(1-alfa_h)*r_h);

            I_sunlit_g = alfa_g * (I_b0[T] + I_d0[T]* Fd_g + I_scatt_g);
            I_shaded_g = alfa_g * (I_d0[T]* Fd_g + I_scatt_g);
            I_avg_g    = alfa_g * (I_b0[T] * Fsunlit_g  + I_d0[T] * Fd_g + I_scatt_g);
      }  //if
      else {
        for(N=0; N<NN; N++) {
            Fsunlit[N] = 0;
            I_scatt0[N] = 0;
            I_scatt_inter[N] = 0;
            I_scatt_intra[N] = 0;
            I_sunlit[N]      = 0;
            I_shaded[N]      = 0;
        }  //loop N

        Fsunlit_h = 0;
        I_scatt_h  = 0;
        I_sunlit_h = 0;
        I_shaded_h = 0;

        Fsunlit_g  = 0;
        I_scatt_g  = 0;
        I_sunlit_g = 0;
        I_shaded_g = 0;
        I_avg_g    = 0;
      }  //else

//===========output
    fprintf(fpt_out, "%04.1lf  %4.1lf %8.1lf %8.1lf %8.1lf ", Time[T], B[T]*180/Pi, I_b0[T]+I_d0[T],
                      I_b0[T], I_d0[T]);
      for(N=0; N<NN; N++) {
        fprintf(fpt_out, "%8.4lf %8.1lf %8.1lf ", Fsunlit[N], I_sunlit[N], I_shaded[N]);
```

```
      }
      fprintf(fpt_out, "%8.4lf %8.1lf %8.1lf ", Fsunlit_h, I_sunlit_h, I_shaded_h);
      fprintf(fpt_out, "%8.4lf %8.1lf %8.1lf %8.1lf\n", Fsunlit_g, I_sunlit_g, I_shaded_g, I_avg_g);

//-------------------------------------------
// Users can use the above output variables to calculate photosynthesis, evapotranspiration
// and energy balance
//-------------------------------------------
  }  //loop T
}

//=====================================================
void Input_Check(int NN, double *Density, double *HH, double *hh, double *DD)
{
  int i,j, IF_Voverlap;
  double P[NNmax], Psum;

  for(i=0; i<NN; i++)   {
      P[i] = DD[i]*DD[i]*Density[i];
      if(P[i]>1.000001) {
        printf("Stratum %d is impossible (P=%lf)\n", i+1, P[i]);
        exit(1);
      }
  }  // loop i

  for(i=0; i<NN; i++) {
      Psum=0;
      for(j=0; j<NN; j++) {
        if(hh[j]>=HH[i] || HH[j]<=hh[i]) IF_Voverlap = 0; //no vertical overlap
        else {
            IF_Voverlap = 1;
            Psum += P[j];
        }
      }  //for j
      if(Psum>1.000001) {
            printf("Strata %d is impossible (Psum=%lf)\n", i+1, Psum);
            exit(1);
      }
```

```
  }  // loop i
}

//=====================================================
void Fraction_Overlap(int NN, double *HH, double *hh, double *Foverlap[])
{
  int j, l;

  for(j=0; j<NN; j++) {
    for(l=0; l<NN; l++) {
        if(j==l) Foverlap[j][l] = 1;
        else if(hh[l]>=HH[j] || HH[l]<=hh[j])
          Foverlap[j][l] = 0; //no overlap vertically
        else if (hh[l]>=hh[j] && hh[l]<=HH[j]) {
            if(HH[l]>=HH[j]) Foverlap[j][l]=(HH[j]-hh[l])/(HH[j]-hh[j]);
        else             Foverlap[j][l]=(HH[l]-hh[l])/(HH[j]-hh[j]);
        }
        else {  // (HH[l]>=hh[j] && HH[l]<=HH[j])
             if(hh[l]>=hh[j]) Foverlap[j][l]=(HH[l]-hh[l])/(HH[j]-hh[j]);
             else             Foverlap[j][l]=(HH[l]-hh[j])/(HH[j]-hh[j]);
        }
      }  // loop l
  }  //loop j
}

//=====================================================
double Calculate_Sunlit_Frac(int NN, double B, int I0, double *Foverlap[], double *K,
         double *Density, double *HH, double *hh, double *DD,
         double *Leaf_area, double *LA_density, double *F1a_ijB, double *fsa_i_B)
{
  int j, k, l, NZ;
  int Mmax, M[NNmax], Steps_Z;

  double H0, H1[NNmax], H2[NNmax];
  double F, F1_ij, F1_i, F_sunlit, LA_sunlit, LA_sunlit0, fsa_i_B0,
           Kp[NNmax], SL[NNmax], SL0, dZ, Z, dA;
  double Lz, sinB, cosB, tanB;
  double Xj1[NNmax], Xjk, Pt, pj[NNmax];
```

```
  H0 = HH[I0] - hh[I0];
  dZ = H0*0.01;

  sinB = sin(B);
  cosB = cos(B);
  tanB = sinB/cosB;

  SL0 = H0/tanB;

  dA = dZ * DD[I0] * cosB;
  Steps_Z = (int) ( (H0+DD[I0]*tanB)/dZ + 0.5);

  for(j=0; j<NN; j++) {
      pj[j] = DD[j]*DD[j]*Density[j];
      Kp[j] = LA_density[j] * K[j];
      H1[j] = HH[j] - hh[I0];
      H2[j] = hh[j] - hh[I0];
      if(H2[j]<0)  H2[j] = 0;
  }

  for(j=0; j<NN; j++) {
      SL[j] = H1[j]/tanB;

      Pt=0;
      for(l=0; l<NN; l++) {
        Pt += pj[l] * Foverlap[j][l];
      }
      if(Pt>1) Pt = 1;

      Xj1[j] = DD[j]*(0.5 - 0.5*Pt + Foverlap[I0][j]);
  }
//---------
  for(j=0; j<NN; j++) {
      Mmax = (int) ((Max_XX-Xj1[j])/(DD[j]) + 1 + 0.5);
      M[j] = Mmax;

      for(k=0; k<Mmax; k++) {
```

```
        Xjk = Xj1[j] + DD[j] * k;

        F = Xjk - DD[j];
        if(SL[j]<F) {
          M[j] = k;
            break;
        }  //if
      }    // loop k
  }      //loopj

  LA_sunlit = 0;
  for(j=0; j<NN; j++) F1a_ijB[j] = 0;
  fsa_i_B0 = 0;
//============
  if(SL0 < DD[I0]) {
    for(NZ=0; NZ<Steps_Z; NZ++) {
      Z = (NZ+0.5)*dZ;

        if(Z<=H0)                Lz = Z/sinB;
        else if (Z<=DD[I0]*tanB) Lz = H0/sinB;
        else                     Lz = DD[I0]/cosB - (Z-H0)/sinB;

        LA_sunlit0 = 1/K[I0] * (1 - exp(-Kp[I0] * Lz)) * dA;

        F1_i =1;
        for(j=0; j<NN; j++) {
          F1_ij = 1;
          if(H1[j]>0) {
              for(k=0; k<M[j]; k++) {
                Xjk = Xj1[j] + DD[j] * k;
                  F = Shading_effects(Z, H1[j], H2[j], DD[j], SL[j], Kp[j],
                                        Xjk, pj[j], sinB, cosB, tanB);
                  F1_ij *= F;
              }  //loop k
            }  //if H1[j]>0
            F1_i *= F1_ij;
            F1a_ijB[j] += F1_ij;
        }  //loop j
```

```
      LA_sunlit += F1_i * LA_sunlit0;
      fsa_i_B0 += (1-F1_i)*(1 - exp(-Kp[I0] * Lz));
    }  //loop NZ
} //if
else { //if SL > DD
    for(NZ=0; NZ<Steps_Z; NZ++) {
      Z = (NZ+0.5)*dZ;

      if(Z<=DD[I0]*tanB) Lz = Z/sinB;
      else if (Z<=H0)    Lz = DD[I0]/cosB;
      else               Lz = DD[I0]/cosB - (Z-H0)/sinB;

      LA_sunlit0 = 1/K[I0] * (1 - exp(-Kp[I0] * Lz)) * dA;

      F1_i = 1;
      for(j=0; j<NN; j++) {
        F1_ij = 1;
        if(H1[j]>0) {
            for(k=0; k<M[j]; k++) {
              Xjk = Xj1[j] + DD[j] * k;
              F = Shading_effects(Z, H1[j], H2[j], DD[j], SL[j], Kp[j],
                                  Xjk, pj[j], sinB, cosB, tanB);
              F1_ij *= F;
            }  // loop k
          }  //if
          F1_i *= F1_ij;
          F1a_ijB[j] += F1_ij;
      }  // loop j
      LA_sunlit += F1_i * LA_sunlit0;

      fsa_i_B0 += (1-F1_i)* (1 - exp(-Kp[I0] * Lz));
    }  //loop NZ
}  //else

fsa_i_B[I0] = fsa_i_B0/Steps_Z;
for(j=0; j<NN; j++) F1a_ijB[j] /= Steps_Z;
F_sunlit = LA_sunlit/Leaf_area[I0];
```

```
  return F_sunlit;
}

//=====================================================
double Shading_effects(double Z, double H1, double H2, double DD, double SL,
      double Kp, double Distance, double pj, double sinB, double cosB, double tanB)
{
  double Z1, Lz, fi_jk;

  fi_jk = 1;

  Z1 = Z + Distance*tanB;
  if(Z1>H2 && Z1<DD*tanB +H1) {  //shade reached
      if(SL<DD) {
        if(Z1<=H1)              Lz = (Z1-H2)/sinB;
        else if(Z1<=H2+DD*tanB) Lz = (H1-H2)/sinB;
        else                    Lz = DD/cosB-(Z1-H1)/sinB;
      }
      else {
        if(Z1<=H2+DD*tanB) Lz = (Z1-H2)/sinB;
        else if(Z1<=H1)    Lz = DD/cosB;
        else               Lz = DD/cosB-(Z1-H1)/sinB;
      }

      fi_jk = (1-pj) + exp(-Kp*Lz)*pj;
  }  //if
  return fi_jk;
}

//=====================================================
double DayLength_SunElevation(int TT, double Lat, double DOY, double *Time,
                                double *Sun_Altitude, double *Sun_RiseSet)
{
// Calculate daily solar elevation, day-length, based on latitude and DOY, and time.
  int i;
  double Day_Length, RAD, DEC, SIN, COS, Hr_Angle, Sin_Z;
  double F, AOB;
```

```
  RAD=Pi/180.0;
  F = sin(23.45*RAD) * cos(2.*Pi*(DOY+10.0)/365.);
  DEC = -asin(F);

  SIN = sin (RAD*Lat)* sin (DEC);
  COS = cos (RAD*Lat)* cos (DEC);
  AOB = SIN/COS;
  if      (AOB<-1)  Day_Length = 0;    //polar night
  else if (AOB>1)   Day_Length = 24.0; //polar day
  else {
      F = asin(AOB);  //normal day and night
      Day_Length = 12.0 * (1 + 2*F/Pi);
  }  //else

  Sun_RiseSet[0] = 12.0 - Day_Length/2.0;
  Sun_RiseSet[1] = 12.0 + Day_Length/2.0;

  for(i=0; i<TT; i++) {
      if(Time[i]>Sun_RiseSet[0] && Time[i]<Sun_RiseSet[1]) {
        Hr_Angle = Pi*(12.0-Time[i])/12.0;
        Sin_Z = SIN + COS * cos(Hr_Angle);
        Sun_Altitude[i] = asin(Sin_Z);
      }  //if
    else {
        Sun_Altitude[i] = 0.00001;
      }
  }  //loop i
  return Day_Length;
}

//=====================================================
double Interpolate(double *XX, double *YY, double X)
{
  int i,j;
  double Y;

  if(X<XX[0]) Y = YY[0]*X/XX[0];
  else if(X>XX[BB-1]) {
```

```
      i = BB-1;
      j = BB-2;
      Y = YY[i] + (YY[j]-YY[i])/(XX[j]-XX[i])*(X-XX[i]);
  }
  else {
      for(i=0; i<BB-1; i++) {
        if(X>=XX[i] && X<=XX[i+1]) {
          j = i+1;
            Y = YY[i]+(YY[j]-YY[i])/(XX[j]-XX[i])*(X-XX[i]);
          break;
        }  //if
      }  //loop i
  }  //else

  return Y;
}
```

--------------------------------The End of the code---

An example of input data file

Users can save the following input dataset into a text file (using the name ‘Input.txt’ and put it directly under the code folder).
The note lines and the notes following the numbers are for easy editing the input file. They are part of the input format and have to be included. A line-by-line explanation of the input data can be found in User’s Manual (Supplement 2).

--------------------------------Following is an example of input data file -----------------------------------

```
An example of input data file for the IPR model (v1.0)
68.5  :Latitude of the site (in degrees)
1.0   :LAI of the herb stratum
0.8   :absorption coefficient of the herb stratum
1.0   :clumping index of the herb stratum
0.9   :absorption coefficient of the ground (1-albedo)
2     :the number of woody strata (max: 5)
10.0 5.0    :top heights of the woody strata (m)
2.0  1.0    :bottom height of the woody strata (m)
1.0  1.0    :crown widths of  the woody strata (m)
3.0  2.0    :local LAI of the woody strata (m2/m2)
0.2  0.2    :plant density of the woody strata (plants/m2)
0.8  0.8    :absorption coefficient of the woody strata
1.0  1.0    :clumping index of the woody strata
197   :day of year (1 - 366)
24    :number of solar radiation data for the day
Time Total_Radiation(W/m2) Fraction_diffuse_radiation
0.0   0.0   1.000
1.0   10.0  0.800
2.0   34.4  0.617
3.0   84.7  0.419
4.0   129.7 0.285
5.0   212.8 0.225
6.0   343.6 0.197
7.0   455.6 0.176
8.0   542.2 0.138
9.0   569.4 0.126
10.0  616.7 0.105
11.0  648.1 0.101
```

```
12.0  645.6 0.097
13.0  581.1 0.109
14.0  538.1 0.117
15.0  505.8 0.147
16.0  425.3 0.172
17.0  330.8 0.193
18.0  242.5 0.212
19.0  170.8 0.240
20.0  94.2  0.294
21.0  40.6  0.365
22.0  10.0  0.514
23.0  0.0   1.000
```

----------------------------------End of the input data file--------------------------------------

The output using the example input data

The output file (with the name ‘Output.txt’) is directly in the folder of the code.
Column 1 (Time): the local time
Column 2 (B): the altitude of the sun (in degrees)
Column 3 (I_Total): total solar radiation on a horizontal surface above the canopy (W/m^2) (model input).
Column 4 (I_Direct): direct solar radiation above the canopy on a horizontal surface (W/m^2) (model input).
Column 5 (I_Diffuse): diffuse solar radiation above the canopy on a horizontal surface (W/m^2) (model input).
There are three columns for each woody stratum and for the herb stratum:
    Column F_Sunlit: the fraction of sunlit leaves of the stratum.
    Column I_Sunlit: the solar radiation absorbed by the sunlit leaves of the stratum (W/m^2 leaf).
    Column I_Shaded: the solar radiation absorbed by the shaded leaves of the stratum (W/m^2 leaf).
There are four columns for the ground:
    Column F_Sunlit: the fraction of sunlit area of the ground.
    Column I_Sunlit: the solar radiation absorbed in the sunlit area of the ground (W/m^2 ground).
    Column I_Shaded: the solar radiation absorbed in the shaded area of the ground (W/m^2 ground).
    Column I_Ground_Avg: the average solar radiation absorbed on the ground (W/m^2 ground).

--------------------------The output using the example input data------------------------------

```
Time    B   I_Total I_Direct I_Diffuse Woody_Stratum_1            Woody_Stratum_2            Herb_Stratum               Ground
----   ---  ------- -------- --------  F_Sunlit I_Sunlit I_Shaded F_Sunlit I_Sunlit I_Shaded Fsunlit I_Sunlit I_Shaded  Fsunlit I_Sunlit I_Shaded I_Ground_Avg
00.0   0.0      0.0      0.0      0.0   0.0000      0.0      0.0   0.0000      0.0      0.0   0.0000      0.0      0.0   0.0000      0.0      0.0      0.0
01.0   0.5     10.0      2.0      8.0   0.0267    105.6      5.1   0.0012    103.8      3.2   0.0000      1.6      1.6   0.0000      3.6      1.8      1.8
02.0   2.4     34.4     13.2     21.2   0.1412    140.9     15.7   0.0065    134.0      8.7   0.0000      4.4      4.4   0.0000     16.8      5.0      5.0
03.0   5.5     84.7     49.2     35.5   0.2799    235.8     31.5   0.0590    221.1     16.8   0.0022      8.0      7.6   0.0002     52.8      8.6      8.6
04.0   9.6    129.7     92.7     37.0   0.4229    261.5     39.5   0.1473    243.7     21.6   0.0192     12.6      8.3   0.0045     92.9      9.5      9.8
05.0  14.4    212.8    164.9     47.9   0.5315    319.3     54.8   0.2619    298.0     33.5   0.0630     28.0     11.3   0.0212    161.7     13.3     16.4
06.0  19.7    343.6    275.9     67.7   0.6071    404.2     77.5   0.3618    378.3     51.6   0.1265     57.8     16.5   0.0561    268.3     20.0     34.0
07.0  25.2    455.6    375.4     80.2   0.6586    443.5     91.2   0.4386    416.4     64.0   0.1926     87.9     20.0   0.1017    362.9     25.0     59.4
08.0  30.6    542.2    467.4     74.8   0.6932    460.1     92.5   0.4943    435.1     67.5   0.2509    112.0     19.8   0.1477    446.4     25.7     87.9
09.0  35.4    569.4    497.7     71.7   0.7154    432.6     89.0   0.5328    410.1     66.5   0.2971    121.4     19.4   0.1872    473.6     25.7    109.5
10.0  39.3    616.7    551.9     64.8   0.7288    435.0     86.6   0.5577    414.5     66.0   0.3300    133.4     18.4   0.2168    521.8     25.0    132.7
11.0  41.9    648.1    582.6     65.5   0.7361    436.6     87.5   0.5719    416.4     67.4   0.3500    140.9     18.7   0.2351    550.1     25.7    149.0
12.0  42.8    645.6    583.0     62.6   0.7383    428.5     85.2   0.5765    409.1     65.9   0.3566    140.6     18.1   0.2413    549.8     25.1    151.7
13.0  41.9    581.1    517.8     63.3   0.7361    391.0     80.8   0.5719    372.1     61.9   0.3500    126.2     17.7   0.2351    490.0     24.1    133.6
14.0  39.3    538.1    475.1     63.0   0.7288    378.7     78.7   0.5577    359.7     59.7   0.3300    116.3     17.3   0.2168    450.8     23.2    115.9
15.0  35.4    505.8    431.4     74.4   0.7154    382.1     84.2   0.5328    360.5     62.5   0.2971    107.7     19.2   0.1872    413.3     25.0     97.7
16.0  30.6    425.3    352.1     73.2   0.6932    356.4     79.4   0.4943    334.6     57.6   0.2509     87.8     18.3   0.1477    340.2     23.2     70.0
```

```
17.0  25.2     330.8     267.0      63.8   0.6586     319.3      68.8   0.4386     298.8      48.3   0.1926      63.8      15.6   0.1017     259.6      19.4      43.8
18.0  19.7     242.5     191.1      51.4   0.6071     282.5      56.3   0.3618     263.8      37.5   0.1265      40.9      12.3   0.0561     186.9      14.9      24.6
19.0  14.4     170.8     129.8      41.0   0.5315     253.2      45.1   0.2619     235.9      27.7   0.0630      22.7       9.5   0.0212     128.0      11.2      13.7
20.0   9.6      94.2      66.5      27.7   0.4229     188.2      29.0   0.1473     175.2      16.0   0.0192       9.3       6.2   0.0045      66.9       7.1       7.3
21.0   5.5      40.6      25.8      14.8   0.2799     121.4      14.3   0.0590     114.3       7.3   0.0022       3.5       3.2   0.0002      26.8       3.6       3.6
22.0   2.4      10.0       4.9       5.1   0.1412      50.4       4.2   0.0065      48.3       2.1   0.0000       1.1       1.1   0.0000       5.6       1.2       1.2
23.0   0.5       0.0       0.0       0.0   0.0000       0.0       0.0   0.0000       0.0       0.0   0.0000       0.0       0.0   0.0000       0.0       0.0       0.0
```

--------------------------The end of the output -----------------------------

Supplement 2

IPR 1.0: An efficient method for calculating solar radiation absorbed by individual plants in sparse heterogeneous woody plant communities

A User's Manual

(Version 1.0, November 2013)

Yu Zhang, Wenjun Chen, and Junhua Li

Canada Centre for Mapping and Earth Observation,
Natural Resources Canada, Ottawa, Canada
Correspondence to: Y. Zhang (yu.zhang@nrcan.gc.ca)

1. The code of the IPR model (1.0)

This user's manual is corresponding to the code of the IPR model (see Supplement 1). The code was developed and tested using Microsoft Visual C++ (Win32 Console Applications).

The code was developed to calculate the solar radiation absorbed by individual plants of a plant community at different times in a day. The input data include the latitude of the location, the day of the year, the conditions of the herb and woody plant strata, and the solar radiation (direct and diffuse) on a horizontal surface above the plant canopy at different time in the day. The outputs include the fractions of sunlit leaf area of the woody plant strata and the herb stratum, the solar radiation of the sunlit and shaded leaves. The model also calculates the fraction of sunlit area of the ground, and the solar radiation in the sunlit and shaded ground areas, and the average solar radiation of the ground. Users can use these outputs to calculate photosynthesis, transpiration and energy balance of the individual plants and the energy balance of the ground.

2. The input file

For simplicity, the input and output files are put directly under the code folder. Users can modify the file path to put the input/output files in different folders. Figure 1 shows an example of input data file. Some brief explanations are included as part of the input file for easy preparation of the data. Following is a line-by-line explanation of the input data.

Line 1: A line of note about the data (1 to 150 characters)

Line 2: Latitude (in degree) of the site. Format: A float followed by a space or tab then a note (1 to 150 characters). Such a note has to be included for corrected reading.

Line 3: Leaf area index of the herb stratum. 0.0 means no herb stratum. In this case, the following two lines (lines 4 and 5) are still needed but will be read as note lines. Format: A float followed by a space or tab then a string (1 to 150 characters).

Line 4: Absorption coefficient of the leaves of the herb stratum (≥ 0.0 m^2/m^2). Format: A float followed by a space or tab then a string (1 to 150 characters).

Line 5: The clumping index of the herb stratum (>0.0). It equals 1 when leaves are randomly distributed. It is larger than 1 when leaves are distributed side by side, and it is smaller than 1 when leaves are stacked on each other. Format: A float followed by a space or tab then a string (1 to 150 characters).

Line 6: The absorption coefficient of the ground (from 0 to 1). It equals 1 minus the albedo. Format: A float followed by a space or tab then a note (1 to 150 characters).

Line 7: The number of woody strata (from 0 to 5, represented by NN). The maximum number of strata is defined as 5 in the code. 0 means no woody strata. In this case, the following seven lines (lines 8 – 14) are still needed but will be read as note lines. Format: An integer followed by a space or tab then a string (1 to 150 characters).

Line 8: The heights of the top of the crowns of the woody strata (m). Format: floats (the number of floats equals the number of woody strata NN) followed by a space or tab then a string (1 to 150 characters).

Line 9: The heights of the bottom of the crowns of the woody strata (m). Format: floats (the number of floats = NN) followed by a space or tab then a string (1 to 150 characters).

Line 10: The widths of the crowns of the woody strata (m). Format: floats (the number of floats = NN) followed by a space or tab then a string (1 to 150 characters).

Line 11: The local leaf area index (LAI) of the crowns of the woody strata (m^2/m^2). The local LAI is the ratio between the leaf area and the ground area directly projected under the crown (m^2 leaf/m^2 land). Format: floats (the number of floats = NN) followed by a space or tab then a string (1 to 150 characters).

Line 12: The density of the plants of the woody strata (plants/m^2). Format: floats (the number of floats = NN) followed by a space or tab then a string (1 to 150 characters).

Line 13: The absorption coefficients of the leaves of the woody strata (0 to 1). Format: floats (the number of floats = N) followed by a space or tab then a string (1 to 150 characters).

Line 14: The clumping indices of the woody strata (>0). They equal 1 when leaves are randomly distributed. They are larger than 1 when leaves are distributed side by side, and are smaller than 1 when leaves are stacked on each other. Format: floats (the number of floats = NN) followed by a space or tab then a string (1 to 150 characters).

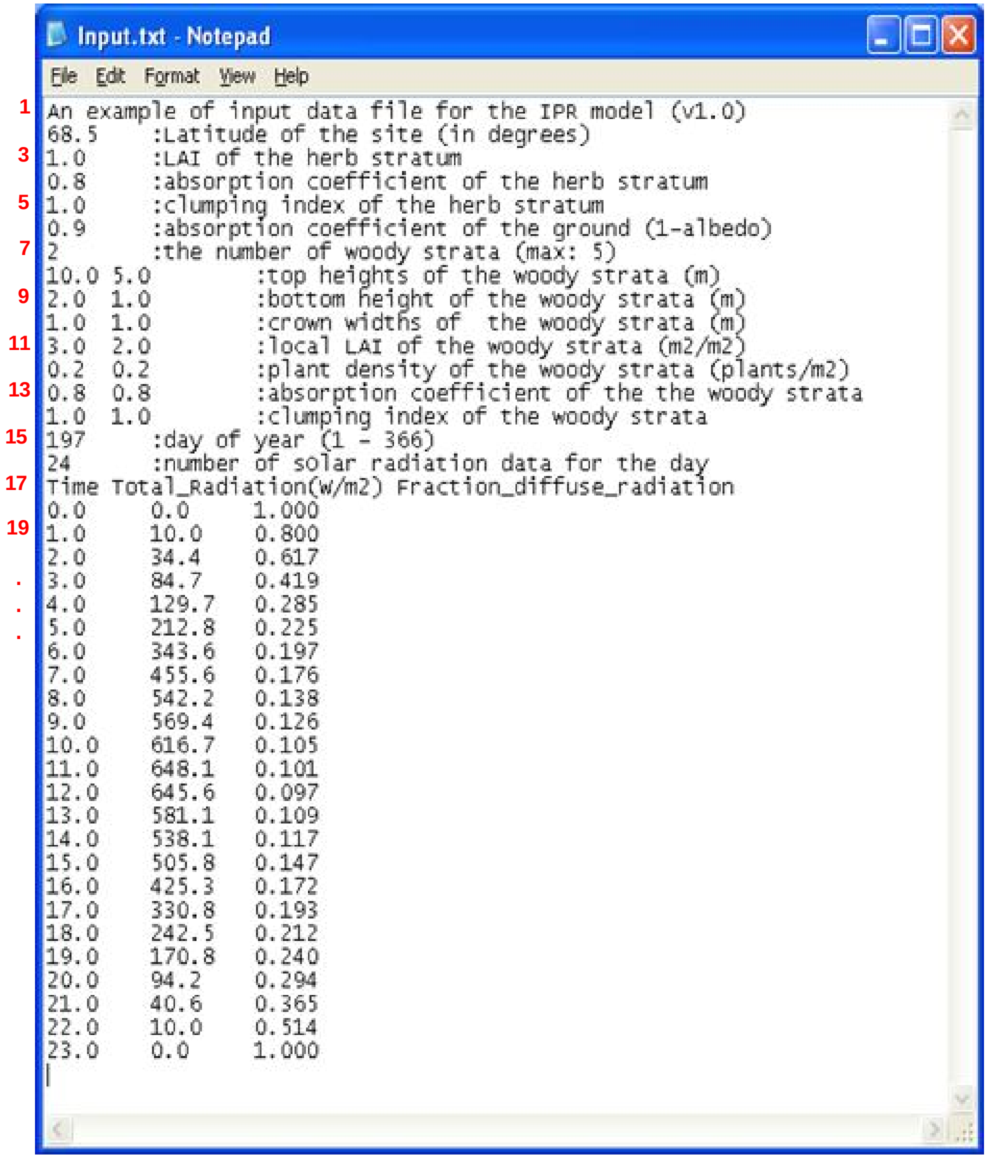
Line 15: The day of the year (1 – 366). Format: An integer followed by a space or tab then a string (1 to 150 characters).

Line 16: The number of radiation observations in the day (1 to 48). Format: An integer followed by a space or tab then a string (1 to 150 characters).

Line 17: a note line (1 to 150 characters)

Line 18: The local time (≥0 but <24), the total solar radiation at that time in a horizontal surface above the canopy (>0), and the fraction of diffuse radiation at that time (0 to 1). The unit of the solar radiation is usually W/m^2, but it can be others. The unit of the calculated canopy and ground solar radiation will be the same as the unit of the input solar radiation data. Format: three floats separated by a space or tab. No string after the three numbers.

Line 19 or more: Same format as line 18. The total number of lines equals the number of radiation observations in the day, defined in the line 16.



```
An example of input data file for the IPR model (v1.0)
68.5     :Latitude of the site (in degrees)
1.0      :LAI of the herb stratum
0.8      :absorption coefficient of the herb stratum
1.0      :clumping index of the herb stratum
0.9      :absorption coefficient of the ground (1-albedo)
2        :the number of woody strata (max: 5)
10.0 5.0        :top heights of the woody strata (m)
2.0  1.0        :bottom height of the woody strata (m)
1.0  1.0        :crown widths of  the woody strata (m)
3.0  2.0        :local LAI of the woody strata (m2/m2)
0.2  0.2        :plant density of the woody strata (plants/m2)
0.8  0.8        :absorption coefficient of the the woody strata
1.0  1.0        :clumping index of the woody strata
197      :day of year (1 - 366)
24       :number of solar radiation data for the day
Time Total_Radiation(w/m2) Fraction_diffuse_radiation
0.0      0.0      1.000
1.0      10.0     0.800
2.0      34.4     0.617
3.0      84.7     0.419
4.0      129.7    0.285
5.0      212.8    0.225
6.0      343.6    0.197
7.0      455.6    0.176
8.0      542.2    0.138
9.0      569.4    0.126
10.0     616.7    0.105
11.0     648.1    0.101
12.0     645.6    0.097
13.0     581.1    0.109
14.0     538.1    0.117
15.0     505.8    0.147
16.0     425.3    0.172
17.0     330.8    0.193
18.0     242.5    0.212
19.0     170.8    0.240
20.0     94.2     0.294
21.0     40.6     0.365
22.0     10.0     0.514
23.0     0.0      1.000
```

Figure 1. An example of input data for the IPR model (The location and solar radiation are based on a clear day on July 16, 1970 in Inuvik, Northwest Territories, Canada).

3. The output file

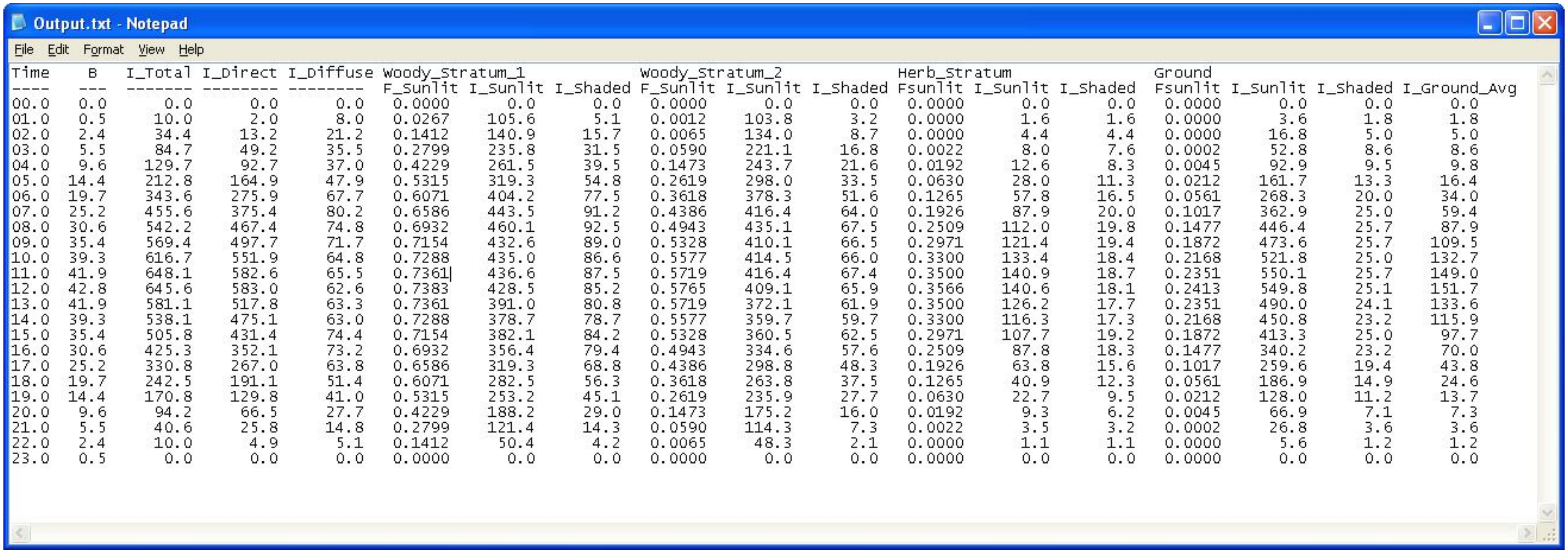
Figure 2 shows the output of the IPR model calculated using the sample input file shown in Fig.1.
Column 1 (Time): the local time (directly from model input).
Column 2 (B): the altitude of the sun (in degrees)
Column 3 (I_Total): total solar radiation in a horizontal surface above the canopy (W/m^2) (from model input).
Column 4 (I_Direct): direct solar radiation above the canopy in a horizontal surface (W/m^2) (from model input).
Column 5 (I_Diffuse): diffuse solar radiation above the canopy in a horizontal surface (W/m^2) (from model input).

There are three columns for each woody stratum and for the herb stratum.
Column F_Sunlit: the fraction of sunlit leaves of the stratum.
Column I_Sunlit: the solar radiation absorbed by the sunlit leaves of the stratum (W/m^2 leaf).
Column I_Shaded: the solar radiation absorbed by the shaded leaves of the stratum (W/m^2 leaf).

There are four columns for the ground:
Column F_Sunlit: the fraction of sunlit area of the ground.
Column I_Sunlit: the solar radiation absorbed on the sunlit area of the ground (W/m^2 ground).
Column I_Shaded: the solar radiation absorbed on the shaded area of the ground (W/m^2 ground).
Column I_Ground_Avg: the average solar radiation absorbed on the ground (W/m^2 ground).

The output file can be opened directly as a text file (Fig. 2) or using Microsoft Office Excel with Fixed Width.

Figure 3 shows the modeled diurnal patterns of the solar radiation conditions of the three strata and the ground.



```
Time     B   I_Total  I_Direct I_Diffuse  Woody_Stratum_1               Woody_Stratum_2                Herb_Stratum                  Ground
----   ---   -------  -------- --------   F_Sunlit  I_Sunlit  I_Shaded  F_Sunlit  I_Sunlit  I_Shaded   Fsunlit  I_Sunlit  I_Shaded   Fsunlit  I_Sunlit  I_Shaded I_Ground_Avg
00.0   0.0       0.0       0.0       0.0    0.0000       0.0       0.0    0.0000       0.0       0.0    0.0000       0.0       0.0    0.0000       0.0       0.0       0.0
01.0   0.5      10.0       2.0       8.0    0.0267     105.6       5.1    0.0012     103.8       3.2    0.0000       1.6       1.6    0.0000       3.6       1.8       1.8
02.0   2.4      34.4      13.2      21.2    0.1412     140.9      15.7    0.0065     134.0       8.7    0.0000       4.4       4.4    0.0000      16.8       5.0       5.0
03.0   5.5      84.7      49.2      35.5    0.2799     235.8      31.5    0.0590     221.1      16.8    0.0022       8.0       7.6    0.0002      52.8       8.6       8.6
04.0   9.6     129.7      92.7      37.0    0.4229     261.5      39.5    0.1473     243.7      21.6    0.0192      12.6       8.3    0.0045      92.9       9.5       9.8
05.0  14.4     212.8     164.9      47.9    0.5315     319.3      54.8    0.2619     298.0      33.5    0.0630      28.0      11.3    0.0212     161.7      13.3      16.4
06.0  19.7     343.6     275.9      67.7    0.6071     404.2      77.5    0.3618     378.3      51.6    0.1265      57.8      16.5    0.0561     268.3      20.0      34.0
07.0  25.2     455.6     375.4      80.2    0.6586     443.5      91.2    0.4386     416.4      64.0    0.1926      87.9      20.0    0.1017     362.9      25.0      59.4
08.0  30.6     542.2     467.4      74.8    0.6932     460.1      92.5    0.4943     435.1      67.5    0.2509     112.0      19.8    0.1477     446.4      25.7      87.9
09.0  35.4     569.4     497.7      71.7    0.7154     432.6      89.0    0.5328     410.1      66.5    0.2971     121.4      19.4    0.1872     473.6      25.7     109.5
10.0  39.3     616.7     551.9      64.8    0.7288     435.0      86.6    0.5577     414.5      66.0    0.3300     133.4      18.4    0.2168     521.8      25.0     132.7
11.0  41.9     648.1     582.6      65.5    0.7361     436.6      87.5    0.5719     416.4      67.4    0.3500     140.9      18.7    0.2351     550.1      25.7     149.0
12.0  42.8     645.6     583.0      62.6    0.7383     428.5      85.2    0.5765     409.1      65.9    0.3566     140.6      18.1    0.2413     549.8      25.1     151.7
13.0  41.9     581.1     517.8      63.3    0.7361     391.0      80.8    0.5719     372.1      61.9    0.3500     126.2      17.7    0.2351     490.0      24.1     133.6
14.0  39.3     538.1     475.1      63.0    0.7288     378.7      78.7    0.5577     359.7      59.7    0.3300     116.3      17.3    0.2168     450.8      23.2     115.9
15.0  35.4     505.8     431.4      74.4    0.7154     382.1      84.2    0.5328     360.5      62.5    0.2971     107.7      19.2    0.1872     413.3      25.0      97.7
16.0  30.6     425.3     352.1      73.2    0.6932     356.4      79.4    0.4943     334.6      57.6    0.2509      87.8      18.3    0.1477     340.2      23.2      70.0
17.0  25.2     330.8     267.0      63.8    0.6586     319.3      68.8    0.4386     298.8      48.3    0.1926      63.8      15.6    0.1017     259.6      19.4      43.8
18.0  19.7     242.5     191.1      51.4    0.6071     282.5      56.3    0.3618     263.8      37.5    0.1265      40.9      12.3    0.0561     186.9      14.9      24.6
19.0  14.4     170.8     129.8      41.0    0.5315     253.2      45.1    0.2619     235.9      27.7    0.0630      22.7       9.5    0.0212     128.0      11.2      13.7
20.0   9.6      94.2      66.5      27.7    0.4229     188.2      29.0    0.1473     175.2      16.0    0.0192       9.3       6.2    0.0045      66.9       7.1       7.3
21.0   5.5      40.6      25.8      14.8    0.2799     121.4      14.3    0.0590     114.3       7.3    0.0022       3.5       3.2    0.0002      26.8       3.6       3.6
22.0   2.4      10.0       4.9       5.1    0.1412      50.4       4.2    0.0065      48.3       2.1    0.0000       1.1       1.1    0.0000       5.6       1.2       1.2
23.0   0.5       0.0       0.0       0.0    0.0000       0.0       0.0    0.0000       0.0       0.0    0.0000       0.0       0.0    0.0000       0.0       0.0       0.0
```

Figure 2. The output of the IPR model (using the input data file shown in Fig.1)

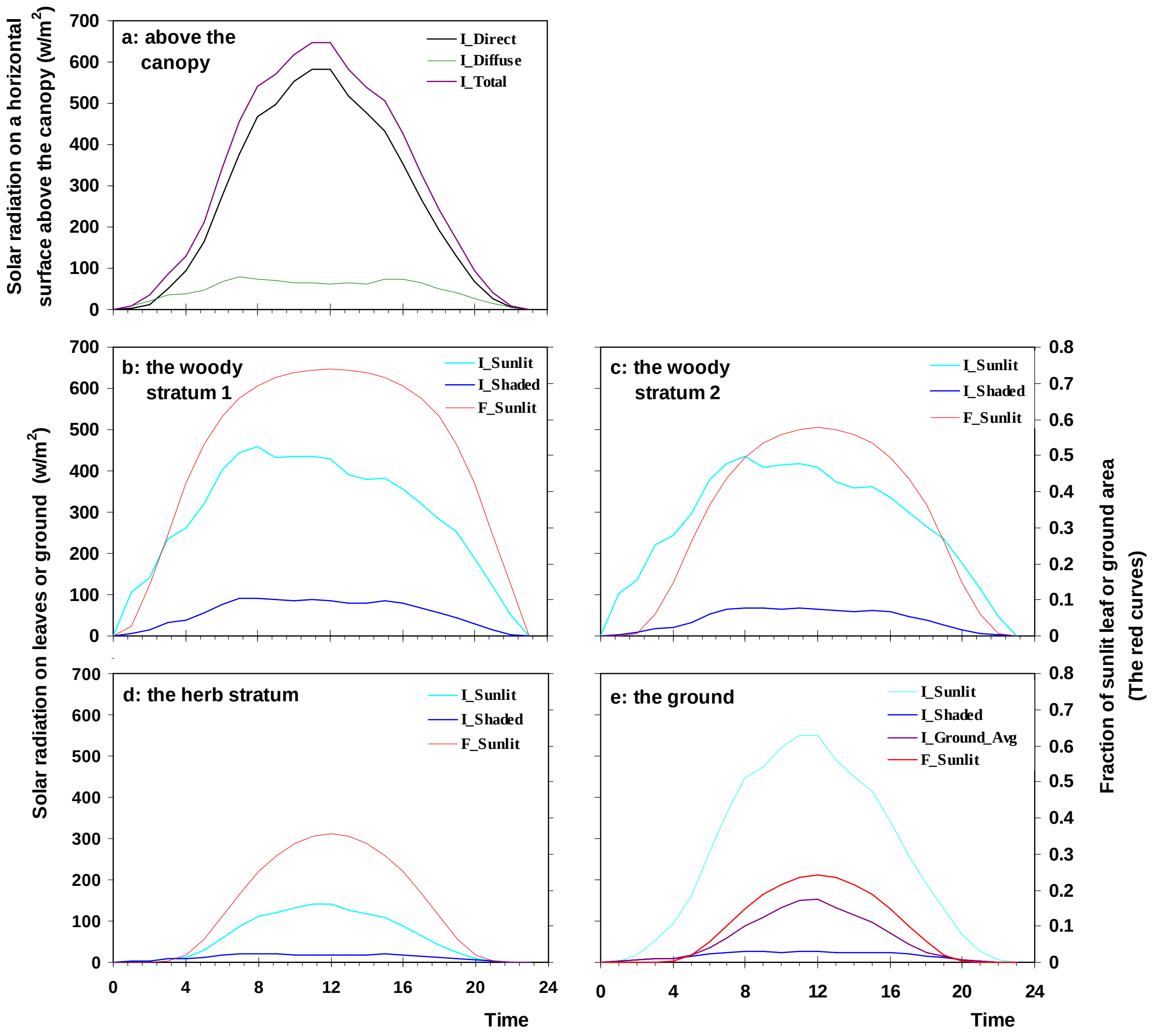


Figure 3. The diurnal patterns of the solar radiation of the plants and the ground (see more explanations in next page)

Figure 3. Panel a: direct, diffuse and total solar radiation on a horizontal surface above the canopy. They were directly from mode inputs. Panels b to d: the fractions of sunlit leaves (the red curves) and the solar radiation of the sunlit (the cyan curves) and shaded leaves (the blue curves) for the two woody strata (b and c) and the herb stratum (d), respectively. Panel e: the fraction of sunlit area on the ground (the red curves) and the solar radiation of the sunlit (the cyan curve) and shaded ground (the blue curve), and the average solar radiation of the ground (the brown curve).

4. Potential applications

The model can be used to estimate the solar radiation absorbed by individual plants in heterogeneous woody plant communities, such as the trees in sparse plant communities, and to estimate the solar radiation on the ground. More importantly, the model can be used as a module of an ecosystem model. The input of the plant conditions and solar radiation above the canopy can be passed from the main model. On each simulation day, the IPR can be called to calculate the solar radiation of different plant strata and the ground, then photosynthesis, evapotranspiration and energy balance can be calculated.