ARPS-CANOPY-DEVS-FIRE (ACFIRE) V1.0 User Manual

DR. MICHAEL T. KIEFER
MICHIGAN STATE UNIVERSITY, EAST LANSING, MICHIGAN, 48823
LAST UPDATED: 7 NOVEMBER 2025

1. ARPS-CANOPY 3
1.1 Background 3
1.2 ARPS-CANOPY formulation 3
1.2.1 Standard ARPS equations 3
1.2.2 ARPS-CANOPY terms and equations 4
1.2.2.1 Momentum and TKE 4
1.2.2.2 Canopy heating/cooling and ground shading 5
1.2.2.3 Canopy-atmosphere moisture exchange 6
1.2.3 Modifications made to ARPS source code 7
1.3 Namelist options 10
1.4 Known issues and limitations 11
2. DEVS-FIRE 13
2.1 Background 13
2.2 DEVS-FIRE heat flux equations 13
2.3 Set up the Project 14
2.4 Run the program 14
2.4.1 Sample fire simulation (without ARPS) 14
2.4.2 Coupled simulation with original ARPS 14
2.4.3 Sample Output 15
2.5 Known issues and limitations 15
3. ARPS-CANOPY-DEVS-FIRE 16
3.1 Background 16
3.2 Configuration 16
3.3 Running AC_{FIRE} 22
3.4 Known issues and limitations 23

1. ARPS-CANOPY

1.1 Background

The development of ARPS-CANOPY occurred as part of a broader project to develop a new smoke dispersion prediction system specifically designed for application to prescribed burns, with a high-resolution numerical model as the meteorological driver. In order to predict smoke dispersion within a forest canopy (defined in the context of this document as the entire vegetation layer, including the crown) and the possible transport of smoke through the canopy-atmosphere interface and into the planetary boundary layer, it is essential that the atmospheric numerical model utilized for this purpose include a canopy parameterization. The meteorological driver chosen for this purpose is the Advanced Regional Prediction System (ARPS). ARPS is designed to simulate microscale- through regional-scale flows, making it particularly useful for transport of smoke across multiple scales, and has been validated extensively (e.g., Xue et al. 2000, 2001).

Unfortunately, the standard ARPS formulation lacks the capability to explicitly simulate atmospheric variables within a multi-layer canopy. In the ARPS framework, as with many mesoscale models, the bulk effect of a vegetation canopy on the atmosphere is computed at the surface (skin) level, beneath the lowest model grid point. A modified version of ARPS, termed ARPS-CANOPY, has been developed (Kiefer et al. 2013) to allow for simulation of air flow within a forest canopy, the salient aspects of which are described in Section 1.2.2. ARPS-CANOPY builds on earlier modifications to ARPS made by Sylvain Dupont at the Institut National de la Recherche Agronomique (INRA), modifications that are discussed in detail in Sections 1.2.2 and 1.2.3 as well as in Dupont and Brunet (2008).

1.2 ARPS-CANOPY formulation

1.2.1 Standard ARPS equations

Neglecting for conciseness the equations for pressure and mixing ratio, the original ARPS equations may be expressed as follows:

The momentum equation may be expressed as

$$\bar{\rho}\left(\frac{\partial \tilde{u}_i}{\partial t}+\tilde{u}_j\frac{\partial \tilde{u}_i}{\partial x_j}\right)=-\frac{\partial}{\partial x_i}\left(\tilde{p}^{'}-\alpha_{div}\frac{\partial \bar{\rho}\tilde{u}_j}{\partial x_j}\right)-2\bar{\rho}\omega_j\epsilon_{ijk}(\tilde{u}_k-\overline{u_k})-\bar{\rho}g\left(\frac{\tilde{\theta}^{'}}{\bar{\theta}}-\frac{\tilde{p}^{'}}{\bar{\rho}c_s^2}\right)\delta_{i3}-\frac{\partial \tau_{ij}}{\partial x_j} \quad (1)$$

where the overtilde indicates grid volume-averaged variables. In this equation, t is time, $u_i(u_1=u,u_2=v,u_3=w)$ is the instantaneous velocity component along $x_i(x_1=x,x_2=y,x_3=z)$, $\bar{\rho}$ is the base state air density, p is air pressure, g is the acceleration due to gravity, and θ is potential temperature. Furthermore, δ_{ij} is the Kronecker delta, ϵ_{ijk} the alternating unit tensor, α_{div} a damping coefficient intended to damp acoustic waves, ω_j is the angular velocity of the earth, and c_s is the speed of sound. Variables with prime notation denote deviations from a horizontally homogeneous, time invariant base state, the latter indicated by an overbar.

The terms on the right-hand side of Eq. (1) represent, respectively, the pressure-gradient force term, the Coriolis term, the buoyancy term, and the turbulent mixing term. Note that as in standard ARPS, the Reynolds or sub-grid scale stress tensor, τ_{ij} is modeled through a subgrid-scale (SGS) gradient transport approach, computed as a function of eddy viscosity (ν_t), itself modeled as the product of a stability-dependent length scale and velocity scale [square-root of SGS turbulent kinetic energy (TKE) (e)]. For more details, the reader is referred to Xue et al. (2000).

The conservation equation for SGS TKE (e) in ARPS-CANOPY may be expressed as:

$$\frac{\partial e}{\partial t} + \tilde{u}_j \frac{\partial e}{\partial x_j} = -\tau_{ij} \frac{\partial \tilde{u}_i}{\partial x_j} - \frac{g}{\bar{\theta}} \tau_{i\theta} \delta_{i3} + \frac{\partial}{\partial x_j}\left(2\nu_t \frac{\partial e}{\partial x_j}\right) - C_\epsilon \frac{e^{\frac{3}{2}}}{l} \tag{2}$$

where the terms on the right-hand side of Eq. (2) represent, respectively, shear production, buoyancy production/destruction, turbulent transport, and dissipation. In Eq. (2), $\tau_{i\theta}$ is the SGS heat flux, l is a dissipation length scale, and C_ϵ is a dissipation constant. For more details, the reader is referred to Xue et al. (2000).

The thermodynamic equation can be expressed as

$$\frac{\partial \theta'}{\partial t} = -w \frac{\partial \bar{\theta}}{\partial z} - \vec{u} \bullet \boldsymbol{\nabla} \theta' + \bar{\rho}^{-1} \boldsymbol{\nabla} \bullet \vec{H} + \frac{1}{\bar{\rho} C_p} \frac{\partial R_N}{\partial z} \tag{3}$$

where θ refers to potential temperature, $\bar{()}$ and $()'$ refer to base state (function of height only) and perturbation variables, $\vec{u}$ is the total wind vector, and $\vec{H}$ is the three-dimensional turbulent heat flux. As in standard ARPS, heat flux is computed in ARPS-CANOPY as $\vec{H} = \bar{\rho} K_H (\boldsymbol{\nabla} \theta)$, where $\bar{\rho}$ is base state density and K_H is the thermal turbulent diffusivity. The terms on the right-hand side of Eq. (3) are adiabatic warming/cooling, advection, turbulent mixing, and radiative forcing.

1.2.2 ARPS-CANOPY terms and equations

1.2.2.1 Momentum and TKE

Following Dupont and Brunet (2008), we have added a canopy drag term

$$-\eta \bar{\rho} C_d A_p \tilde{V} \tilde{u}_i \tag{4}$$

to the standard ARPS momentum equation [Eq. (1)] to account for drag that occurs due to the presence of the canopy elements. In this term, C_d is the mean drag coefficient of the canopy and A_p ($\mathrm{m^2\ m^{-3}}$) is the plant area density of the vegetation, defined as the one-sided area of all plant material. The magnitude of the resolved-scale velocity, V, is defined as $V = (u^2 + v^2 + w^2)^{\frac{1}{2}}$. A modification has been made to the original term presented in in that a factor of η is included to incorporate effects of vegetation fraction less than unity, following the work of Yamada (1982) and Sun et al. (2006). We have introduced η, a parameter that represents the fraction of a grid cell covered by trees, to account for the fact that ARPS-CANOPY is designed to be run with grid cells large enough that an assumption of land-cover homogeneity across the grid cell is not

necessarily appropriate. The A_p profile is considered to be representative of the canopy density within the vegetated portion of each grid cell.

Also following Dupont and Brunet (2008), a turbulence sink term

$$-2\eta C_d A_p \tilde{V} e \tag{5}$$

was added to the SGS TKE equation [Eq. (2)] in order to account for the loss of SGS TKE to both heat and very small (and thus dissipative) wake-scale eddies, a process often referred to as a "short-circuit" of the inertial eddy cascade (Raupach and Thom 1981; Finnigan 2000).

Following Kanda and Hino (1994), we have also added a turbulence production term

$$\beta\eta C_d A_p \tilde{V}^3 \tag{6}$$

to the SGS TKE equation [Eq. (2)] to represent the production of SGS TKE in the wakes of canopy elements, at scales large enough that the turbulence does not dissipate immediately yet small enough that it remains unresolved. The coefficient β represents the fraction of kinetic energy lost due to canopy drag that contributes to wake production in SGS flow. A value of 0 means that no kinetic energy lost from the resolved-scale flow due to canopy drag transfers to wake-scale turbulence (i.e., energy is lost to heat only), whereas a value of 1 means that all kinetic energy lost from the resolved-scale flow due to canopy drag goes to the production of wake-scale turbulence.

1.2.2.2 Canopy heating/cooling and ground shading

In addition to the modification of the momentum and TKE equations, changes have been made to the thermodynamic equation, Eq. (3) to account for the heat source/sink in the canopy and the shading of the ground surface.

We follow Sun et al. (2006) and compute net radiation flux at canopy top (at height h) as

$$R_{Nh} = (1 - \alpha_t)S + \varepsilon_c(R_{Lh}\downarrow - R_{Lh}\uparrow) \tag{7}$$

where α_t is the canopy albedo, S represents the incoming solar radiation flux intercepting the top of the canopy, ε_c is canopy emissivity, and $R_{Lh}\uparrow$ and $R_{Lh}\downarrow$ are upward and downward longwave radiation. The formulation of Eq. (7) is otherwise identical to the standard ARPS ground radiation budget, except that here we use a constant value of albedo appropriate for forested areas, and the outgoing longwave component ($R_{Lh}\uparrow$) is computed as a function of air temperature at canopy top, rather than skin temperature.

Following Sun et al. (2006), we have also prescribed a profile of net radiation that produces an approximately exponential decay within the canopy,

$$R_{Np}(z) = R_{Nh}\left[exp\{-kP_L(z)\} - \eta\left(1 - \frac{z}{h}\right)exp\{-kP_L(0)\}\right] \tag{8}$$

In Eq. (8), k is an extinction coefficient, and $P_L(z) = \int_z^h A_p\,(z)dz$ is the local plant area index (PAI), which indicates the plant area per unit horizontal area of the canopy above height z. Equation (8) states that the transmission of net radiation through a vegetation canopy exhibits an approximately exponential decay with increasing penetration depth into the canopy, as a function of the local PAI.

With the net radiation inside the canopy computed, we replace the last term in Eq. (3) with

$$\frac{(1-\eta)}{\rho_a C_p}\frac{\partial R_N}{\partial z} + \frac{\eta}{\rho_a C_p + \rho_c C_c}\left(1+\frac{1}{B}\right)^{-1}\frac{\partial R_{Np}}{\partial z} \tag{9}$$

where θ is the potential temperature of the air and R_N is the net radiation flux within the clearing fraction of each grid box. Equation (9) states that the time rate of change of potential temperature inside the canopy is computed as the weighted sum of vertical radiation flux divergence in the clearing fraction of each grid cell, and vertical radiation flux divergence in the vegetated part of each grid cell. The leading factor in the second term on the right-hand side of Eq. (9) accounts for heat storage in the canopy elements (through the canopy element volumetric heat capacity, $\rho_c C_c$), as well as partitioning of energy into sensible and latent heat flux (through the Bowen ratio, B).

Lastly, the net radiation budget at the ground is given by:

$$R_{NG} = \eta R_{Nh} exp[-kP_L(0)] + (1-\eta)[(1-\alpha_G)S + \varepsilon_G(R_{LG}\downarrow - R_{LG}\uparrow)] \tag{10}$$

where symbols with subscript "G" refer to ground surface equivalents of the canopy top parameters in Eq. (7), and $P_L(0)$ is local PAI computed at the ground (i.e., total PAI). Note that Eq. (10) is used by the land surface model as part of the integration of skin temperature whereas the net radiation flux in Eq. (8) is used to compute the canopy source term in the thermodynamic equation [Eq. (9)].

1.2.2.3 Canopy-atmosphere moisture exchange

The moisture scheme in ARPS-CANOPY utilizes the existing canopy evapotranspiration parameterization (Noilhan and Planton 1989; Pleim and Xiu 1995) that is a standard element of the ARPS land-surface model to compute a source term for the specific humidity equation. The evapotranspiration and canopy water evaporation terms in the ARPS land-surface model are expressed as follows:

$$E_{et} = \eta\rho(1-F_w)[q_{vsat}(T_s) - q_v](R_a + R_s)^{-1} \tag{11}$$

$$E_v = \eta\rho F_w[q_{vsat}(T_s) - q_v]R_a^{-1} \tag{12}$$

where F_w is the wet fraction of the canopy $\left[F_w = (W_v/W_{vmax})^{2/3}\right]$, W_v is the canopy water content (depth; specified in namelist), T_s is the upper-layer soil temperature, q_v is atmospheric specific humidity, and R_a and R_s are aerodynamic and surface resistances, respectively. The aerodynamic resistance is parameterized as $R_a = \left(C_{dq}V\right)^{-1}$, where C_{dq} is the moisture exchange coefficient and V is the wind speed one-half grid point below the lowest atmospheric grid level.

In ARPS-CANOPY, we take the sum of Eqs. (11) and (12), termed E_{can}, and distribute this canopy evapotranspiration/liquid evaporation term according to the plant area density at each grid level inside the canopy, i.e.,

$$E_{can3d}(x,y,z) = E_{can}(x,y)\left(\frac{A_p(z)}{\sum_{z=0}^{z=h} A_p\,(z)}\right) \tag{13}$$

Finally, the canopy evaporation source term is computed as

$$q_{vcanfrc} = \frac{E_{can3d}}{dz} \tag{14}$$

where dz is the vertical grid spacing, and the source term is combined with the other source terms in the ARPS specific humidity equation. Note that multiplying E_{can3d} by $1/dz$ in Eq. (14) is equivalent to dividing by volume air density (volume of air in grid cell divided by horizontal area of cell). This formulation is broadly similar to the moisture scheme in the rural and urban-canopy model introduced into MM5 by Dupont et al. (2004), however, the evaporation terms [i.e., Eqs. (11) and (12)] in that model are computed at every model level and the resistance terms are height dependent in the canopy.

1.2.3 Modifications made to ARPS source code

ARPS-CANOPY contains a number of subroutines not originally in ARPS that were developed by Sylvain Dupont:

- **drag_force** in force3d.f90. This subroutine computes the canopy drag term [Eq. (4)] and outputs 3D forcing arrays (*uforce*,*vforce*,*wforce*) which are passed to the parent subroutine (**frcuvw**) and added to the contributions from the other (non-canopy) forcing terms for use in computing the three wind components (u,v,w) at the next timestep.
- **initcanopee** in initlib3d.f90. This subroutine defines the canopy at the initial timestep. In this subroutine, one 2D array is created, *kcanopee* (uppermost grid level in canopy), and a series of 3D arrays are also generated, including *af_veg* (plant area density), *cd_veg* (canopy drag coefficient), *as_veg* (vertical distribution of net radiation in canopy), and *ah_veg* (horizontal canopy area density; this parameter was added by Sylvain Dupont for use with a particle dispersion code that is not part of ARPS-CANOPY).
- **wake_tke** in tke3d.f90. This subroutine computes the subgrid-scale turbulence sink term [Eq. (5)] and closely resembles **drag_force** in terms of the way the code is organized. As part of the development of ARPS-CANOPY, this subroutine was modified from the version that Sylvain Dupont developed to also compute the wake production term [Eq. (6)]. Similar to the subroutine **drag_force**, this subroutine outputs a 3D forcing array (*tkeforce*) that is used by the parent subroutine (**solvtke**) to integrate TKE forward in time.

The following existing ARPS subroutines were modified by Sylvain Dupont (all modifications are denoted in subroutines by commented lines with 'SYLVAIN' or 'sylvain'):

- **arps.f90**. The program was modified to initialize/allocate canopy-related arrays and pass them to various subroutines.
- **binreadsplit** in binio3d.f90. A write statement was modified (very minor).
- **extbdt** in exbc3d.f90. A minor change was made to the subroutine that reads in predicted variables from external boundary files.
- **frcuvw** in force3d.f90. This subroutine was modified to add a call to **drag_force**.
- **inibase** in inibase3d.f90. Parameters *lvlprof* and *depthp*, defined at the beginning of the subroutine for use in interpolating unevenly spaced sounding data to the model grid, were modified slightly. This is only relevant for applications where a single sounding is used to initialize the model.

- **initial** in init3d.f90. The subroutine was modified to pass canopy-related variables to **initgrdvar**.
- **initgrdvar** in initlib3d.f90. This subroutine was modified in two ways. First, it was modified to add a call to **initcanopee**; second, additional variables were added to an existing call to the subroutine **rstin**, for use with a particle dispersion sub-model (not part of ARPS-CANOPY).
- **chkstab**,**initout**, and **output** in out3d.f90. Subroutines **initout** and **chkstab** were modified to enable processing of canopy-related arrays, but nothing else was modified (it remains at this time unfinished). Subroutine **output** was modified to pass along particle dispersion related arrays to subroutine **rstjoinout**, and minor changes were made to a number of IF/THEN statements that control the frequency of history and restart file data dumps.
- **rstin**,**rstout**,**rstinsplit**, and **rstjoinout** in rst3d.f90. Subroutine **rstout** and **rstjoinout** were modified to output particle dispersion model-related arrays to restart files (former: serial version, latter: parallel version). Similarly, **rstin** and **rstinsplit** were modified to read in particle dispersion model-related arrays from restart files.
- **pbldepth** and **sfcphysics** in sfcphy3d.f90. Portions of subroutine **pbldepth** were modified to ensure that PBL depth is computed properly. In standard ARPS, a search is conducted starting from the lowest model level, defining the top of the PBL as the level where virtual potential temperature equals the value at the lowest grid level. Sylvain Dupont made a small but important modification to the subroutine to begin the search at the top of the canopy instead of the lowest grid level. Without this modification, daytime stable layers inside the canopy would result in extremely shallow PBL depths. Subroutine **sfcphysics** was modified slightly to include canopy-related parameters in the call to subroutine **pbldepth**.
- **cordintg** and **tinteg** in tinteg3d.f90. These two subroutines, which orchestrate the computation of forcing terms and the subsequent integration of the prognostic variables, were modified to include canopy (and particle dispersion) variables in each subroutine as well as calls to other subroutines.
- **solvtke** in tke3d.f90. A call to **wake_tke** was added to subroutine **solvtke**; the output of subroutine **wake_tke**, *tkeforce*, is subsequently amended as the output of the various forcing terms (e.g., shear production) are summed. Also, various subroutines for a particle dispersion sub-model (not actually part of ARPS-CANOPY), (**aleat,aleatoire,concatint,cpyary1d**, **cpyary1digtslpfn,particles, ponder,sourceps,source1ps**), were added to tke3d.f90.

During the development of ARPS-CANOPY various existing ARPS subroutines were modified (all modifications denoted in subroutines by commented lines with ‘MTK’):

- **radiation** in radfrc3d.f90. Several changes were made in this subroutine. First a temporary 2D array *temp2d* is defined to store air temperature at canopy top, except for grid points with no (or extremely sparse) forest cover, where skin temperature is stored instead. This array is then passed to the existing subroutine **radtrns** where outgoing longwave radiation flux at canopy top (or ground surface in clearing points) is computed. Second, albedo is hardwired to 0.1 in grid points with forest cover, and 0.3 in clearing points, following . Note, for both modifications described here, grid points with very sparse canopies (PAI less than about 0.37) are treated as vegetation-free points. Lastly,

relevant canopy parameters (e.g., PAI) are passed to **radtrns** for further radiation calculations.

- **radtrns** in radtrns3d.f90. Changes made in this subroutine constitute the heart of the canopy heating/cooling additions to ARPS and are contained at the end of the subroutine, after the model has computed radiation flux divergence (hereafter, radiation forcing) without canopy effects. The additional code added to the subroutine computes the thermodynamic forcing term associated with radiation absorbed/emitted from the canopy. The radiation forcing term computed earlier in the subroutine (array *radfrc*) is redefined as the background, or clearing-fraction radiation forcing (array *radfrc_bg*), and has nothing to do with the canopy itself. In order to compute the radiation forcing in the vegetated portion of each grid cell, i.e., the heating/cooling of the air that results from the heating/cooling of the canopy elements, the factor $\left\{\frac{\eta}{\rho_a C_p+\rho_c C_c}\left(1+\frac{1}{B}\right)^{-1}\right\}$ in Eq. (9) is computed. In the process, several canopy parameters are computed or hardwired including canopy biomass, canopy density (ρ_c), specific heat of the canopy (C_c), and canopy Bowen Ratio (B). At the end of the modified section, radiation forcing is computed as a weighted sum of the background forcing in the clearing fraction of the grid cell, and the canopy radiative forcing in the vegetated fraction. If the grid cell is devoid of vegetation, the radiation forcing defaults to the background value.
- **soilebm_frc** in soilebm3d.f90. In general, all vegetation in standard ARPS is contained below the lowest model level. In the standard version of **soilebm_frc**, the ground heat capacity at each grid point is computed as a weighted sum of the soil heat capacity (in the clearing fraction of the grid cell) and vegetation heat capacity (in the vegetated fraction). In ARPS-CANOPY, however, since model grid points are embedded within the vegetation, the vegetation heat capacity directly influences air temperature through C_c in Eq. (9). Thus, we make two assumptions: (1) ground heat capacity equals soil heat capacity regardless of vegetation fraction, and (2) there is no vegetation below the lowest model grid level (the latter assumption avoids the need to account for vegetation heat capacity when computing soil temperature tendency).
- **sfcphysics** in sfcphy3d.f90. Modifications to this subroutine are restricted to the incorporation of Eq. (10), i.e., the shading effect of the canopy. An array *temxy1* is added that contains the weighted sum of the reduced net radiation flux in the vegetated fraction of the grid cell, and the unadulterated net radiation flux in the clearing fraction of the grid cell. If a grid cell is completely devoid of trees, *temxy1* reverts back to the full values stored in the array *rnflx*. After the calculation, *temxy1* is passed along to **soilebm** or **ousoil**, depending on the choice of soil model (2-layer force restore or 6-layer OU scheme) in the arps.input namelist, and soil temperature tendencies are computed.
- **cftmix** in tmix3d.f90. A known issue with ARPS (standard and canopy versions) is under-mixing of scalars when fine vertical grid spacing is utilized near the surface [see Section 1.5 for more details]. Because the explicit simulation of turbulent structures within a forest canopy requires the use of fine vertical grid spacing, the length scale issue can have a pronounced impact on ARPS-CANOPY simulations. The dependence of turbulent length scale (and therefore turbulent mixing) on vertical grid spacing was bypassed to a limited degree in ARPS-CANOPY by setting a minimum vertical length

scale, as well as a minimum eddy viscosity. Thus, in **cftmix**, values of vertical length scale in the array *lenscl* are required to be greater than or equal to 10 m, and values of eddy viscosity in array *kmv* are required to be greater than or equal to 0.1 m^2 s^{-1}.

1.3 Namelist options

All ARPS-CANOPY options are contained in the namelist 'arps_canopy.input' (a modified version of the standard 'arps.input' namelist), in namelist group '&canopy'. If running any programs other than arps_mpi that use the 'arps.input' namelist (e.g., arpssfc, ext2arps), use of 'arps_canopy.input' may cause problems due to the additional namelist entries. The ARPS-CANOPY options are as follows:

- **can_opt**: 0 = no canopy; 1 = canopy. This is the flag for switching ARPS-CANOPY on/off. If **can_opt** is set to '0', then all equations presented in Section 1.2 effectively revert to their non-canopy form. However, an important distinction must be made here. The code itself does not completely revert to the original ARPS source code when the canopy flag is set to '0'; the canopy code remains, but the plant area density is set to zero everywhere. Thus, canopy drag terms in the momentum and TKE equations are computed but are equal to zero, the vertical radiation flux divergence term reverts to the standard ARPS form [i.e., the last term in Eq. (3)], and canopy shading is set to zero.
- **can_input**: 1 = pre-defined canopy shape; 2 = read in from external ascii file.
- **can_shape**: Canopy shape. Profiles from Fig. 1 in Kiefer et al. (2018):

  = *1, Shape A.* Dense crown layer and sparse trunk space; broadly representative of a maritime or loblolly pine stand.

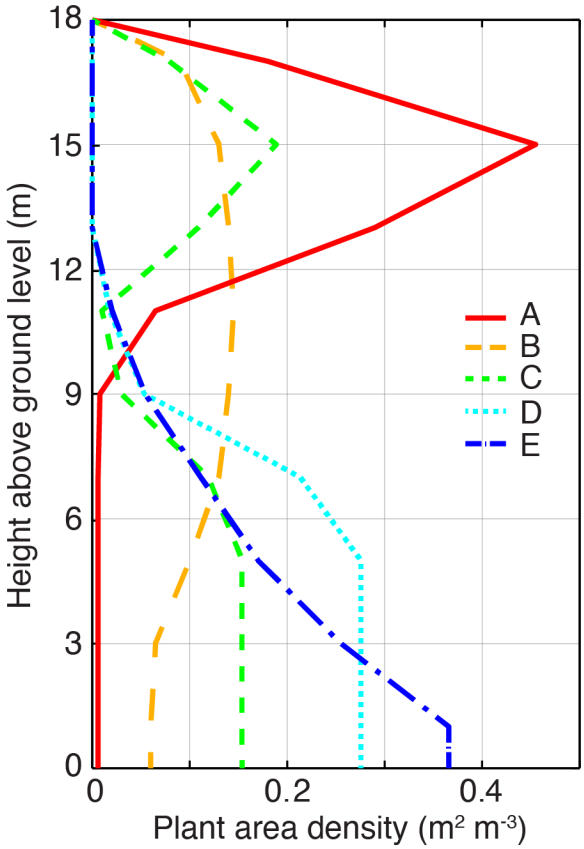
  = *2, Shape B.* More uniformly distributed profile than shape A; based on a deciduous forest stand in Ontario, Canada.

  = *3, Shape C.* Variation on shape A in which two distinct crown and understory vegetation layers exist.

  = *4, Shape D.* Variation on shape C in which the crown vegetation is removed and redistributed to the understory, but the overall shape of the understory profile is retained. May be suitable for wintertime deciduous forests with evergreen understory vegetation.

  = *5, Shape E.* Vegetation density decreases exponentially away from the surface and is based on Light Detection and Ranging (LiDAR)-derived profiles of Ap obtained in a mixed Pitch Pine-Scrub Oak forest stand in New Jersey.

  The shapes are taken directly from Kiefer et al. (2018) and are illustrated in Fig. 1.
- **can_pai**: Plant area index provided by user. This is a real number that can vary from 0 to $< \infty$. At this time, the user is only able to specify a homogeneous canopy in the 'arps_canopy.input' namelist. ARPS-CANOPY can simulate flow through heterogeneous canopies, but the user must modify the subroutine **initcanopee** in initlib3d.f90 to define a heterogeneous canopy (or read in from an external file).
- **can_data**: File name for external ascii file (not supported at this time).



Vertical profiles of plant area density, plotted as a function of normalized canopy height, Z/h. The user can specify which canopy shape to utilize in their ARPS-CANOPY simulation in the '&canopy' namelist group (see Section 1.3 for more details).

1.4 Known issues and limitations

- Under-mixing of potential temperature (and other scalars): A bias is known to exist with ARPS-CANOPY (as well as standard ARPS) in which gradients of temperature near the surface are stronger than suggested by available observations, when relatively fine vertical grid spacing is used. The problem is believed to result from the use of vertical grid spacing to define the vertical mixing scale, which ultimately is used to compute vertical heat flux. ARPS was not originally designed to be applied with vertical grid spacing O(1 m). Note that this issue is also expected to affect other scalars, such as moisture. As a way of working around the use of vertical grid spacing in computing length scales, a minimum length scale is applied in all ARPS-CANOPY simulations (5 m). Also, a minimum eddy viscosity is applied (0.1 $m^2\ s^{-1}$), to ensure that some minimum amount of turbulent mixing occurs in all simulations with ARPS-CANOPY (see the discussion of changes made to subroutine **cftmix** in Section 1.2.3).
- Assumption about equal heating/cooling rate of canopy elements and atmosphere: Since we follow Sun et al. (2006), our model assumes that the rate of heating/cooling of vegetation elements is identical to that of adjacent canopy air spaces. Froelich et al. (2011) argue that such an assumption is not acceptable, with the largest error occurring

near sunrise (sunset) when canopy elements warm (cool) rapidly through radiative gain (loss), and air temperature changes lag behind. The lag is the result of a relatively slow sensible heat exchange between canopy elements and the surrounding air (in contrast, momentum transport from the air to the canopy elements is very rapid). For more details about this problem, see Froelich et al. (2011) and Belcher et al. (2012).

- Hardwired canopy properties (Bowen Ratio, biomass, specific heat, albedo): At this time, several canopy properties are static and cannot be changed without modifying the source code. The values are hardwired in subroutine **radtrns** as such: Bowen ratio β = 1.0; canopy biomass = 4.99 kg m^{-2}; specific heat of canopy C_c = 2760 J kg^{-1} K^{-1}; canopy albedo α_c = 0.15. Such values are expected to apply to a wide variety of cases, but in some cases the values may not be appropriate and the user may need to modify the values in **radtrns**.
- Uncertainty about wake production coefficient [β in Eq. (5)]: The coefficient β represents the fraction of kinetic energy lost due to canopy drag that contributes to wake production in SGS flow. A value of 0 means that no kinetic energy lost from the resolved-scale flow due to canopy drag transfers to wake-scale turbulence (i.e., energy is lost to heat only), whereas a value of 1 means that all kinetic energy lost from the resolved-scale flow due to canopy drag goes to the production of wake-scale turbulence. Unfortunately, there are no studies that we are aware of that have considered how the coefficient varies with LAI or PAI. Following Kanda and Hino (1994), we set the wake production coefficient to 0.1.
- Drag coefficient not a function of x, y, z, or t: We follow Dupont and Brunet (2008) and use a constant coefficient (C_d = 0.2), while acknowledging that in reality C_d may decrease as wind speeds increase due to streamlining effects (Rudnicki et al. 2004), and conversely, C_d may increase as wind speeds decrease due to the greater role of molecular viscosity at weak wind speeds. It is worth noting that while use of a constant drag coefficient is a simplification of a complex process, it is not without precedent, having been applied to canopies with a wide range of canopy densities (e.g., Shaw and Schumann 1992; Watanabe 2004; Sun et al. 2006; Dupont and Brunet 2008). However, much uncertainty exists with respect to use of a constant drag coefficient of 0.2 for sparse canopies, and caution must be exercised when applying ARPS-CANOPY to areas with thin canopies.
- Coarse horizontal grid spacing (for operational application): Grid spacing is the greatest challenge to applying ARPS-CANOPY to operational prediction, due to the need for fine grid spacing inside the canopy. However, it has been found that ARPS-CANOPY simulations with horizontal grid spacing O(100 m) can be run in near-real time. Sensitivity experiments have been performed with relatively coarse 90-m horizontal grid spacing and were shown to retain the overall mean profile shape and diurnal trends seen in simulations with smaller grid spacing (Kiefer et al. 2013).

2. DEVS-FIRE

2.1 Background

DEVS-FIRE is a raster-based fire behavior model that is a form of cellular automata (CA) model (von Neuman, 1966) utilized in other two-way coupled atmosphere-fire models (e.g. CAWFE, WRF-SFIRE, and QUIC-Fire). Note that DEVS-FIRE is a surface fire model only and is incapable of simulating crown fires; this is an important limitation that must be kept in mind when applying AC_{FIRE}. In DEVS-FIRE, the fire is made up of contiguous cells with fire propagation occurring via cell-to-cell interactions, with spread rates from an ignited cell toward neighboring unburned cells calculated from Rothermel's equations (Rothermel, 1972) with an assumed elliptical spread pattern around each ignited cell based on Finney (1998). Sensible and latent heat fluxes from the fire to the atmosphere are computed using a fuel-model-dependent static fuel loss curve model as implemented in other mesoscale models such as WRF-SFIRE (Mandel et al., 2011) and CAWFE (Coen, 2013).

2.2 DEVS-FIRE heat flux equations

$$\phi_{h=} -\frac{dF}{dt}\frac{1}{1+M}w_n h = \frac{1}{T_f}e^{-\frac{t}{T_f}}\frac{1}{1+M}w_n h \tag{15}$$

$$\phi_{q=} -\frac{dF}{dt}\frac{1}{1+M}w_n h = \frac{1}{T_f}e^{-\frac{t}{T_f}}\frac{0.56+M}{1+M}w_n L \tag{16}$$

$$F(t) = e^{-\frac{t}{T_f}} \tag{17}$$

$$T_f = \frac{W}{0.8514} \tag{18}$$

Equations (15) and (16) correspond to the sensible and latent heat flux equations, respectively, which in turn are functions of the fuel mass fraction (F) [Eq. (17)] and an e-folding timescale (T_f) [Eq. (18)]. The original equations for sensible and latent heat flux [Eqs. (2) and (3) in Dahl et al., 2015] are modified to compute the time derivative of fuel mass fraction, as opposed to the finite difference approximation of the derivative as originally implemented in the model. This modification ensures that the communication interval between ARPS-CANOPY and DEVS-FIRE (the period over which fuel mass fraction loss is computed in DEVS-FIRE) does not unduly influence the resultant heat fluxes (not shown).

2.3 Set up the Project

This is a java project with source code included. The project depends on three jar files (included in the lib directory):

- DESEngine.jar: the discrete event simulation engine
- CA_presentation.jar: the cellular automata presentation package that supports display of simulation results in a 2D grid fashion (cellular automata)
- swt_x86_64.jar (or swt_x86.jar for 32-bit machine): the Standard Widget Toolkit (SWT) used by the graphical user interface (GUI)

Include the jar files in your project (e.g., a project based on Eclipse IDE), or in the classpath when you compile the code from the command line.

The project also includes a resources directory, which includes data files (e.g., fuel, terrain, weather data) that are needed for running the simulation.

2.4 Run the program

2.4.1 Sample fire simulation (without ARPS)

Run desFire.testing.TestFireSystem and one will see the simulation of the September 2000 Moore Branch Fire (section 2.4.3).

2.4.2 Coupled simulation with original ARPS

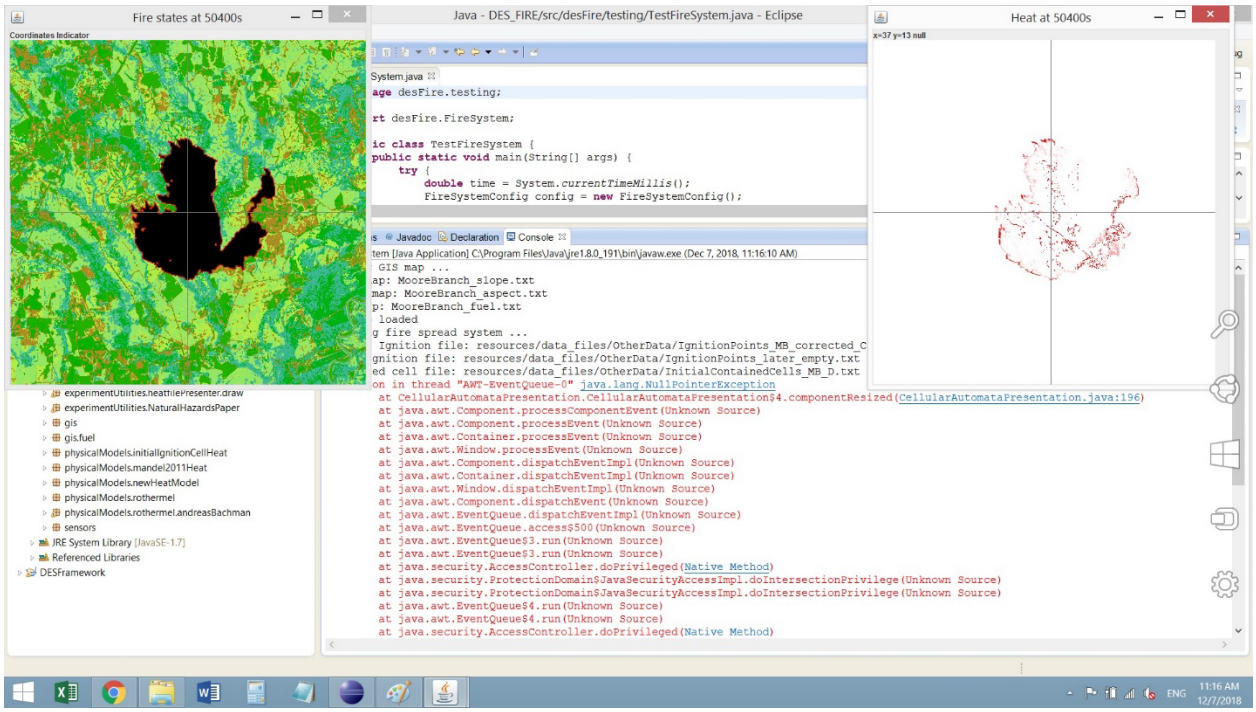
Run desFire.couplingWithARPS.CouplingShell with an argument of the configuration file path.

Configuration file example (in the resources/OtherData directory):

```
#Configuration of DES-FIRE/ARPS coupled simulation
#Tue May 22 17:39:00 EDT 2012

# GIS data
GISFileFolder=GISData/
fuelFileName=MooreBranch_fuel.txt
slopeFileName=MooreBranch_slope.txt
aspectFileName=MooreBranch_aspect.txt

# Initial wind schedule
initialWeatherFileName=weather_artificial_DEVSFIRE.txt

# Ignition
ignitionFile=IgnitionPoints_MB_corrected.txt
laterIgnitionFile=IgnitionPoints_later_empty.txt

# Initial contained cells
initialContainedCells=InitialContainedCells_MB.txt
```

```
# Coupling related data (numberOfSteps is the number of weather update steps)
weatherUpdateInterval=1.0
heatUpdateInterval=1
numberOfSteps=3

# When nested is true, a grid will be modeled by nxn cells, n is the scaleFactor
nested=false
scaleFactor=1

# If use new heat model for initial ignition cells
useNewHeatModelForInitialIgnitionCells = false
```

2.4.3 Sample Output



2.5 Known issues and limitations

DEVS-FIRE is a surface fire model only and is incapable of simulating crown fires. Sensible heat flux is not a function of spread rate (or background wind speed) due to the use of a static fuel loss curve model [Eq. (17)]. Fire spread in DEVS-FIRE has limited degrees of freedom: each ignited cell communicates only with the eight surrounding cells. Backing fire spread is represented indirectly via an elliptical fire spread assumption for each ignited cell.

3. ARPS-CANOPY-DEVS-FIRE

3.1 Background

No current coupled atmosphere-fire modeling framework simultaneously resolves atmospheric responses to wildland fires across scales from hundreds of meters to hundreds of kilometers, incorporates a comprehensive suite of physical parameterizations, and explicitly resolves some scales of atmospheric turbulence within and above a forest canopy. ARPS-CANOPY/DEVS-FIRE (hereafter, AC_{FIRE}) is a new two-way coupled atmosphere-fire model that integrates the canopy-resolving turbulence capability of ARPS-CANOPY (Kiefer et al., 2013) with the dynamic atmosphere-fire coupling of ARPS/DEVS-FIRE (Dahl et al., 2015).

3.2 Configuration

AC_{FIRE} is controlled via two configuration files:

- DEVS-FIRE CouplingConfig ("devs_couplingconfig.txt")
- ARPS-CANOPY namelist ("arpscanopy_namelist.input")

DEVS-FIRE CouplingConfig file parameters:

Parameter	Definition	Possible values	Comment
GISFileFolder	Path to directory containing GIS data files	Character string	Generally set as ~/resources/data_files/GISData/
fuelFileName	Fuel type map	Character string	Values in file must be between 1 and 13 (corresponding to Anderson 1982 fuel models) File is stored in ~/resources/data_files/GISData/
slopeFileName	The angle between the surface and the horizontal plane in the up-slope direction	Character string	Values in file must be between 0 (flat terrain) and 90 (vertical). File is stored in ~/resources/data_files/GISData/
aspectFileName	The down-slope direction	Character string	Values in file must be 0 (flat terrain) and 360 (north).

			File is stored in ~/resources/data_files/GISData/
initialWeatherFileName	Name of file containing 6-m air temperature, wind speed, and wind direction	Character string	This is only used if DEVS-FIRE is run in uncoupled mode (without ARPS). Although the parameter name contains the word "initial", this file is used throughout the DEVS-FIRE simulation. The weather variables can vary with time, at intervals as small as one minute, by including more than one line. File is stored in ~/resources/data_files/OtherData/
ignitionFile	File containing DEVS-FIRE grid cells ignited prior to the start of the simulation.	Pair of non-zero integers, corresponding to i and j grid indices, separated by a tab	File is stored in ~/resources/data_files/DEVS_HeatFiles and ~/resources/data_files/OtherData/
laterIgnitionFile	File containing DEVS-FIRE grid cells manually ignited at some time after ignition	Pair of non-zero integers, corresponding to i and j grid indices, separated by a tab	File is stored in ~/resources/data_files/OtherData/
initialContainedCells	File containing DEVS-FIRE grid cells that the fire can never spread into, i.e., forbidden cells.	Pair of non-zero integers, corresponding to i and j grid indices, separated by a tab	File is stored in ~/resources/data_files/OtherData/
weatherUpdateInterval	Frequency of weather file output from ARPS [seconds]	Non-zero real value	Must be identical to value specified in ARPS namelist ("grdwrite_dt").

heatUpdateInterval	Frequency of heat output from DEVS-FIRE [seconds]	Non-zero real value	Must be identical to value specified in ARPS namelist ("fire_dt").
numberOfSteps	Number of DEVS-FIRE heat output files. The product of "heatUpdateInterval" and "numberOfSteps" is equivalent to the duration of the DEVS-FIRE simulation.	Non-zero integer	The product of "heatUpdateInterval" and "numberOfSteps" must be identical to the "tstop" parameter in the ARPS namelist.
useHeatWriter	Option to output heat files from DEVS-FIRE	Logical: "true" or "false"	Must be set to "true" for two-way coupled ARPS-DEVS-FIRE simulations.
showPresentationLayer	Option to display maps of fire burn progression and sensible heat flux, in pop-up windows, during DEVS-FIRE simulation	Logical: "true" or "false"	Generally speaking, set to "true" if troubleshooting simulation, and "false" otherwise.
heatOutputFolder	Path to directory where heat files are output by DEVS-FIRE and read in by ARPS.	Character string	Must be identical to path specified in ARPS namelist ("fire_dir").
nested	Option to run DEVS-FIRE with two domains, one nested within the other.	Logical: "true" or "false"	This parameter is set to "false" in all "CouplingConfig" files provided by Xiaolin Hu at Georgia State University.
scaleFactor	Number of cells in each direction of nested grid.	Non-zero integer	
waitingTimeout	The longest period of time that DEVS-FIRE will wait for weather output file from ARPS [seconds].	Non-zero real value	
useNewHeatModelForInitialIgnitionCells	This option allows the user to use the "quasi-discrete"	Logical: "true" or "false"	This parameter is set to "false" in all "CouplingConfig" files

	version of DEVS-FIRE (see Dahl et al. 2015)		provided by Xiaolin Hu at Georgia State University

Anderson, H, 1982: Aids to determining fuel models for estimating fire behavior. USDA Forest Service, Intermountain Forest and Range Experiment Station. General Technical Report INT-122.

ARPS-CANOPY namelist parameters:

&initialization namelist block:

Parameter	Definition	Possible values	Comment
firemod	Fire heat source	*0: no fire heat source; 1: ARPS internal heat source (no DEVS-FIRE); 2: DEVS-FIRE heat source (2-way coupled simulation)*	
fingers	Stationary fireline shape	0: straight fireline; 1: sinusoidal fireline	Only used if firemod=1
fire_width	Idealized fireline width [meters]	Non-zero real value	Only used if firemod=1
fire_ymargin	Distance from southern end of fireline to southern lateral boundary [meters]	Non-zero real value	Only used if firemod=1
fingerwidth	North-south width of isolated fire finger (oriented about domain center) [meters]	Non-zero real value	Only used if firemod=1 and fingers=1
fire_amp	East-west amplitude of sinusoidal fireline [meters]	Non-zero real value	Only used if firemod=1 and fingers=1
lfbuff	Option for exponential-damping buffer zone at lateral boundaries	0: no buffer; 1: buffer	Not implemented
idealflux	surface sensible heat flux from idealized stationary fireline [K*kg/(m**2*s)]	Non-zero real value. Negative sign indicates upward flux.	Only used if firemod=1. Heat flux units are the same as ARPS's sensible heat flux variable. Divide by Cp [1004

			m**2/(s**2*K)] to transform units from W/m2 to K*kg/(m**2*s).
fire_dir	Path to directory containing heat output from DEVS-FIRE	Character string	Only used if firemod=2. Must be identical to path specified in DEVS-FIRE CouplingConfig file ("heatOutputFolder"), except with "/HeatmapFiles" omitted.
fire_dt	Time interval between heat output updates from DEVS-FIRE [seconds]	Non-zero real value	Only used if firemod=2. Must be identical to value specified in DEVS-FIRE CouplingConfig file ("heatUpdateInterval").
dxfire	DEVS-FIRE grid spacing [meters]	Non-zero real value	Only used if firemod=2. Must be identical to value specified in DEVS-FIRE GIS data files ("cellsize").
nxfire	Number of DEVS-FIRE grid cells in x-direction	Non-zero integer	Not implemented. This is specified elsewhere: (1) in the DEVS-FIRE GIS data files ("ncols"), and (2) in the ARPS namelist inside the "&output" block ("nwritepts_x").
nyfire	Number of DEVS-FIRE grid cells in y-direction	Non-zero integer	Not implemented. This is specified elsewhere: (1) in the DEVS-FIRE GIS data files ("nrows"), and (2) in the ARPS namelist inside the "&output" block ("nwritepts_y").

&output namelist block:

Parameter	Definition	Possible values	Comment
sfcgridpt_write	Option to write out ARPS surface variables to text files, for	0: do not write out to text files; 1: write out to text files	Only used if firemod=2.

	use by DEVS-FIRE		
gridpt_dirname	Path to directory where ARPS text files are stored, for use by DEVS-FIRE	Character string	Only used if firemod=2. Must be identical to path specified in parameter "f" in "CouplingShell.java" (search for "File f").
gridpt_runname	Filename header for ARPS surface variable text files, for use by DEVS-FIRE	Character string	Only used if firemod=2. Must be identical to string specified in parameter "fileflag" in "CouplingShell.java" (search for "String fileflag").
grdwrite_dt	Time interval at which ARPS surface variables are to be written to text files, for use by DEVS-FIRE [seconds]	Non-zero real value	Only used if firemod=2. Must be identical to value specified in DEVS-FIRE CouplingConfig file ("weatherUpdateInterval").
grdwrite_all	Option to write out ARPS surface variables at all ARPS grid points.	0: Write out data at DEVS-FIRE grid points only (default); 1: Write out data at all ARPS grid points.	Not implemented.
grdwrite_ctr	Option to control how the DEVS-FIRE domain is positioned within the larger ARPS domain.	0: DEVS-FIRE domain is specified using lower-right coordinates of DEVS-FIRE domain ("grdul_lat" and "grdul_lon"); 1: DEVS-FIRE domain is centered within the ARPS *physical* domain (NX-3 x NY-3, where NX and NY are the number of ARPS grid points in the x- and y-directions).	Only used if firemod=2. Dahl et al. (2015) state that option "0" should be used for real case simulations, and option "1" should be used for idealized simulations.
grdul_lat	Latitude at the center of the lower-right corner grid cell in DEVS-FIRE domain [decimal degrees].	Non-zero real value	Only used if firemod=2 and grdwrite_ctr=0. This is the *only* place where the DEVS-FIRE domain is specified in geographic coordinates. Despite use of "ul" in parameter name,

			this *does not* correspond to "upper-left" corner.
grdul_lon	Longitude at the center of the lower-right corner grid cell in DEVS-FIRE domain [decimal degrees].	Non-zero real value	Only used if firemod=2 and grdwrite_ctr=0. This is the *only* place where the DEVS-FIRE domain is specified in geographic coordinates. Despite use of "ul" in parameter name, this *does not* correspond to "upper-left" corner.
grdwritell_i	i-coordinate at which to begin writing out ARPS surface variables to text file, for use by DEVS-FIRE	Non-zero integer	Not implemented (parameter is computed internally in ARPS program; arps.f90).
grdwritell_j	j-coordinate at which to begin writing out ARPS surface variables to text file, for use by DEVS-FIRE	Non-zero integer	Not implemented (parameter is computed internally in ARPS program; arps.f90).
nwritepts_x	Number of DEVS-FIRE grid points in the x-direction.	Non-zero integer	Only used if firemod=2. Must be identical to value specified in the DEVS-FIRE GIS data files ("ncols").
nwritepts_y	Number of DEVS-FIRE grid points in the y-direction.	Non-zero integer	Only used if firemod=2. Must be identical to value specified in the DEVS-FIRE GIS data files ("nrows").

3.3 Running AC$_{FIRE}$

AC$_{FIRE}$ is run using three shell scripts: arpscanopydevs_run.sh (master script), arpscanopy_run.sh, and devs_run.sh. The master script ensures that ARPS-CANOPY and DEVS-FIRE run simultaneously, with the two models communicating over the time interval specified separately in the ARPS-CANOPY namelist (parameters "grdwrite_dt" and "fire_dt") and the DEVS-FIRE CouplingConfig file (parameters "heatUpdateInterval" and "weatherUpdateInterval"). It is strongly recommended that the same time interval be used for all four update interval parameters.

3.4 Known issues and limitations

The forest canopy in ARPS-CANOPY and the fuel bed in DEVS-FIRE are specified independently of each other, with static Ap data utilized during the coupled simulation. One way of interpreting this is that the vegetation in ARPS-CANOPY does not burn; the reader is reminded that DEVS-FIRE, and as a consequence AC_{FIRE}, is incapable of simulating crown fires. Furthermore, use of an ASCII-file messaging system for communication between the atmospheric and fire models ASCII-file messaging system imposes limitations on the computational efficiency of the coupled model.

REFERENCES

Belcher, S. E., I. N. Harman, and J. J. Finnigan, 2012: The wind in the willows: Flows in forest canopies in complex terrain. Annu. Rev. Fluid Mech., 44, 479–504.

Coen, J.L., 2013: Modeling Wildland Fires: A Description of the Coupled Atmosphere-Wildland Fire Environment Model (CAWFE), National Corporation for Atmospheric Research (NCAR) Technical Note NCAR/TN-500+STR, NCAR, https://opensky.ucar.edu/system/files/2024-08/technotes_511.pdf (last access: 3 April 2025).

Dahl, N., H. Xue, X. Hu, and M. Xue, 2015: Coupled Fire–Atmosphere Modeling of Wildland Fire Spread Using DEVS-FIRE and ARPS. Nat. Hazards, 77, 1013–1035.

Dupont, S., and Y. Brunet, 2008: Influence of foliar density profile on canopy flow: A large-eddy simulation study. Agric. For. Meteorol., 148, 976–990.

Dupont, S., T. L. Otte, and J. K. S. Ching, 2004: Simulation of meteorological fields within and above urban and rural canopies with a mesoscale model (mm5). Boundary-Layer Meteorol., 113, 111–158.

Finney, M.A., 1998: FARSITE: Fire Area Simulator-Model Development and Evaluation, Res. Pap. RMRS-RP-4, Rocky Mountain Research Station, https://doi.org/10.2737/RMRS-RP-4.

Finnigan, J. J., 2000: Turbulence in plant canopies. Annu. Rev. Fluid Mech., 32, 519–571.

Froelich, N. J., C. S. B. Grimmond, and H. P. Schmid, 2011: Nocturnal cooling below a forest canopy: Model and evaluation. Agric. For. Meteorol., 151, 957–968.

Kanda, M., and M. Hino, 1994: Organized structures in developing turbulent flow within and above a plant canopy, using a large eddy simulation. Boundary-Layer Meteorol., 68, 237–257.

Kiefer, M. T., S. Zhong, W. E. Heilman, J. J. Charney, and X. Bian, 2013: Evaluation of an ARPS-based canopy flow modeling system for use in future operational smoke prediction efforts. J. Geophys. Res., 118, 6175–6188.

Kiefer, M.T., S. Zhong, W.E. Heilman, J.J. Charney, and X. Bian, 2018: A Numerical Study of Atmospheric Perturbations Induced by Heat from a Wildland Fire: Sensitivity to Vertical Canopy Structure and Heat Source Strength, J. Geophys. Res., 123, 2555–2572.

Mandel, J., Beezley, J.D., and Kochanski, A.K., 2011: Coupled Atmosphere-Wildland Fire Modeling with WRF 3.3 and SFIRE 2011, Geosci. Model Dev., 4, 591–610, https://doi.org/10.5194/gmd-4-591-2011.

Raupach, M., and A. Thom, 1981: Turbulence in and above plant canopies. Annu. Rev. Fluid Mech., 13, 97–129.

Rothermel, R.C., 1972: A Mathematical Model for Predicting Fire Spread in Wildland Fuels, Res. Pap. INT-115, USDA Forest Service, https://www.fs.usda.gov/rm/pubs_int/int_rp115.pdf (last access: 8 April 2025).

Rudnicki, M., S. J. Mitchell, and M. D. Novak, 2004: Wind tunnel measurements of crown streamlining and drag relationships for three conifer species. Can. J. For. Res., 34, 666–676.

Shaw, R. H., and U. Schumann, 1992: Large-eddy simulation of turbulent flow above and within a forest. Boundary-Layer Meteorol., 61, 47–64.

Sun, H., T. L. Clark, R. B. Stull, and T. A. Black, 2006: Two-dimensional simulation of airflow and carbon dioxide transport over a forested mountain. Part I: Interactions between thermally-forced circulations. Agric. For. Meteorol., 140, 338–351.

von Neumann, J., 1966: The Theory of Self-Reproducing Automata, University of Illinois Press.

Watanabe, T., 2004: Large-eddy simulation of coherent turbulence structures associated with scalar ramps over plant canopies. Boundary-Layer Meteorol., 112, 307–341.

Xue, M., K. K. Droegemeier, and V. Wong, 2000: The Advanced Regional Prediction System (ARPS) – A multi-scale nonhydrostatic atmosphere simulation and prediction model. Part I: Model dynamics and verification. Meteorol. Atmos. Phys., 75, 463–485.

Xue, M., K. K. Droegemeier, V. Wong, A. Shapiro, K. Brewster, F. Carr, D. Weber, Y. Liu, and D. Wang, 2001: The Advanced Regional Prediction System (ARPS) – A multi-scale nonhydrostatic atmosphere simulation and prediction tool. Part II: Model physics and applications. Meteorol. Atmos. Phys., 76, 143–165.

Yamada, T., 1982: A numerical study of turbulent airflow in and above a forest canopy. J. Meteorol. Soc. Jpn., 60, 439–454.