Development of Terrestrial Dynamical Cores for E3SM

Gautam Bisht[1]

[1]Climate & Ecosystem Sciences Division
Lawrence Berkeley National Laboratory

2018 Scientific Discovery through Advanced Computing
(SciDAC-4) Principal Investigator Meeting

July 24, 2018

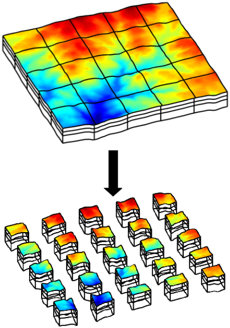
Scientific requirements

Current generation land surface models (LSMs), including ELM, routinely neglect many critical multi-component, multi-physics processes such as:

Scientific requirements

Current generation land surface models (LSMs), including ELM, routinely neglect many critical multi-component, multi-physics processes such as:

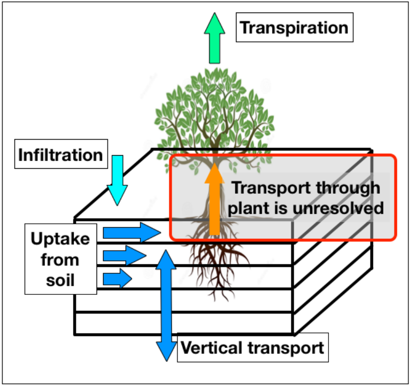
- Lateral redistribution of water, energy, and nutrients



Scientific requirements

Current generation land surface models (LSMs), including ELM, routinely neglect many critical multi-component, multi-physics processes such as:

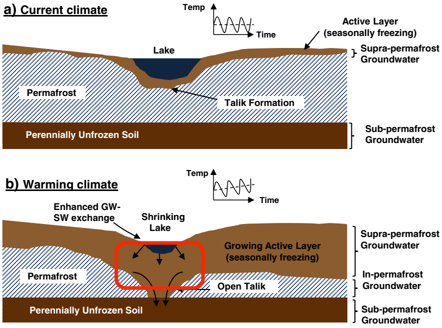
- Lateral redistribution of water, energy, and nutrients
- Transport of water through soil-plant continuum



Scientific requirements

Current generation land surface models (LSMs), including ELM, routinely neglect many critical multi-component, multi-physics processes such as:

- Lateral redistribution of water, energy, and nutrients
- Transport of water through soil-plant continuum
- Advective transport of energy



Kurylyk et al. (2014), Earth-Science Reviews

Computational requirements

E3SM's 10-year vision of a sub-kilometer resolution in terrestrial components imposes several key computation requirements for the terrestrial dynamical core (dycore):

- Scalable solver for nonlinear parabolic PDE with 10^{10} unknowns
- Support unstructured grids
- Spatial discretization that accounts for non-orthogonal grids
- Flexible framework to assemble a tightly coupled multi-component, multi-physics problem
- Runtime configurability to use a range of numerical algorithms

Computational requirements

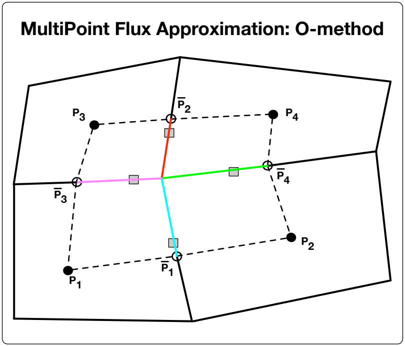
E3SM's 10-year vision of a sub-kilometer resolution in terrestrial components imposes several key computation requirements for the terrestrial dynamical core (dycore):

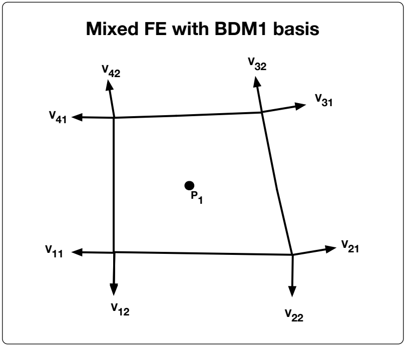
- Scalable solver for nonlinear parabolic PDE with 10^{10} unknowns
- Support unstructured grids
- Spatial discretization that accounts for non-orthogonal grids
- Flexible framework to assemble a tightly coupled multi-component, multi-physics problem
- Runtime configurability to use a range of numerical algorithms

Develop a rigorously verified, spatially adaptive, scalable, multi-physics dycore for global-scale modeling of three-dimensional subsurface processes in E3SM. The dycore will use PETSc to provide numerical solution of discretized equations.

Early results: Spatial discretization

- Identified two spatial discretization methods that account for non-orthogonal grids and have been previously applied to solve for flow and transport processes



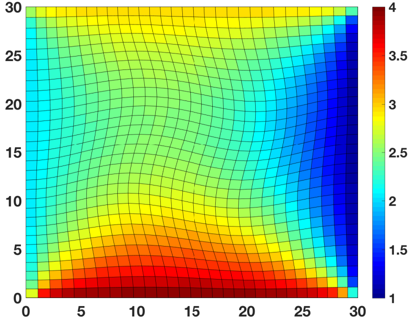


- Both methods lead to similar set of nonlinear equations with unknowns pressure values at cell centers

Early results: Spatial discretization

- Developed a prototype code for solving 2D steady-state diffusion equation using MPFA-O method

$$\nabla \cdot (K \nabla P) = 0 \text{ with}$$
$$K = 1,\ P_{south} = 4,\ P_{north} = 3,\ P_{right} = 1, \text{ and } P_{left} = 2$$

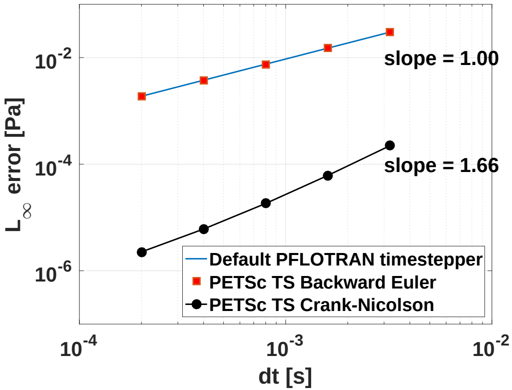


- Preliminary comparison of our results show good agreement with the MATLAB Reservoir Simulation Toolbox

Early results: Temporal discretization

- Implemented PETSc TS–based solver in PFLOTRAN, which uses first-order spatial discretization

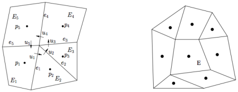
  Problem setup: Evolution of liquid pressure towards a hydrostatic equilibrium starting with homogenous conditions in a 1D soil column



BER–ASCR Partnership

Theory

2092 MARY F. WHEELER AND IVAN YOTOV



A. Five elements sharing a vertex. B. Pressure stencil.

FIG. 2.3. *Interactions of the degrees of freedom in MFMFE.*

2.7. Reduction to a cell-centered stencil. We next describe how the MFMFE method reduces to a system for the pressures at the cell centers. Let us consider any interior vertex $\mathbf{r}$ and suppose that it is shared by k elements $E_1, \dots, E_k$; see Figure 2.3(A) for a specific example with 5 elements. We denote the edges (faces) that share the vertex by $e_1, \dots, e_k$, the velocity basis functions on these edges (faces) that are associated with the vertex by $\mathbf{v}_1, \dots, \mathbf{v}_k$, and the corresponding values of the normal components of $\mathbf{u}_h$ by $u_1, \dots, u_k$. Note that for clarity the normal velocities on Figure 2.3(A) are drawn at a distance from the vertex.

Since the quadrature rule $(K^{-1}\cdot, \cdot)_Q$ localizes the basis functions interaction (see (2.34)–(2.35)), taking $\mathbf{v} = \mathbf{v}_1$ in (2.41), for example, will only lead to coupling u_1 with u_5 and u_2. Similarly, u_2 will only be coupled with u_1 and u_3, etc. Therefore, the k equations obtained from taking $\mathbf{v} = \mathbf{v}_1, \dots, \mathbf{v}_k$ form a linear system for $u_1, \dots, u_k$.

PROPOSITION 2.7. *The $k \times k$ local linear system described above is symmetric and positive definite.*

Proof. The system is obtained by taking $\mathbf{v} = \mathbf{v}_1, \dots, \mathbf{v}_k$ in (2.41). On the left-hand side we have

$$(K^{-1}\mathbf{u}_h, \mathbf{v}_i)_Q = \sum_{j=1}^{k} u_j (K^{-1}\mathbf{v}_j, \mathbf{v}_i)_Q \equiv \sum_{j=1}^{k} a_{ij} u_j, \quad i = 1, \dots, k.$$

Implementation

```
/*
  There will need to be a quadrature for each element type in the
  mesh. Neither is this dim independent. It could be generalized to
  simply use as locations the vertices of the reference element.
 */
#undef __FUNCT__
#define __FUNCT__ "PetscDTWheelerYotovQuadrature"
PetscErrorCode PetscDTWheelerYotovQuadrature(DM dm,AppCtx *user)
{
  PetscFunctionBegin;
  PetscErrorCode ierr;
  ierr = PetscQuadratureCreate(PETSC_COMM_SELF,&(user->q));CHKERRQ(ierr);
  PetscInt dim=2,nq=4;
  PetscReal *x,*w;
  ierr = PetscMalloc1(nq*dim,&x);CHKERRQ(ierr);
  ierr = PetscMalloc1(nq    ,&w);CHKERRQ(ierr);
  x[0] = -1.0; x[1] = -1.0;
  x[2] =  1.0; x[3] = -1.0;
  x[4] = -1.0; x[5] =  1.0;
  x[6] =  1.0; x[7] =  1.0;
  w[0] = 0.25; w[1] = 0.25; w[2] = 0.25; w[3] = 0.25;
  ierr = PetscQuadratureSetData(user->q,dim,1,nq,x,w);CHKERRQ(ierr);
  PetscFunctionReturn(0);
}
```

- Discussions between BER and ASCR colleagues have been extremely useful in translating the mixed FE theory into code
- Application of PETSc's Discretization Technology (DT) capability to mFE discretization is expected to improve DT

Thank you