NSF ITR AP&IM 0205198

COMPUTATIONAL TOOLS DEVELOPMENT

DECEMBER 2002

Adrian Sandu

Dacian Daescu

Gregory R. Carmichael

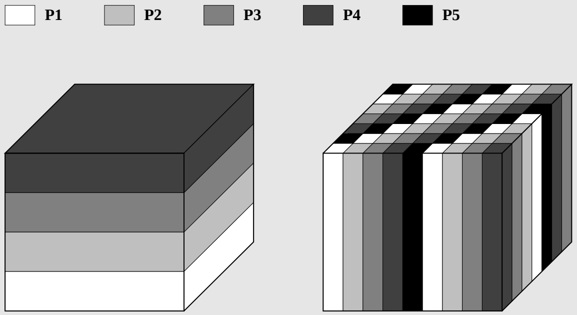
PAQMSG

www.cs.mtu.edu/∼asandu/Software/Parallel

Comm. Library for Parallel. of AQMs on Struct. Grids

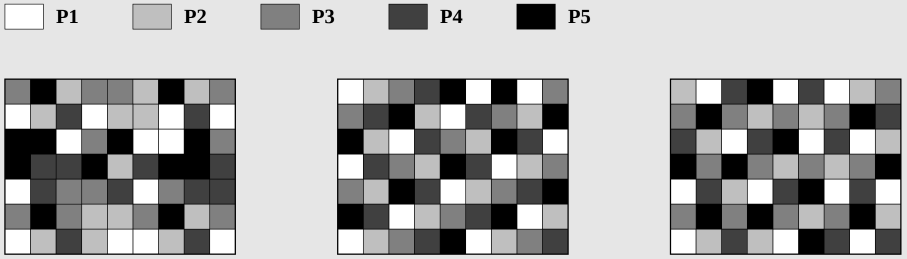
- Fortran90 and MPI (hierarchy of F90 modules);
- Master/worker, domain decomp., static partitioning;
- Data types: 4D, 3D, 2D arrays;
- Alloc (local/global); Distribution; Gathering; Shuffling;

Example Domain Decomp. & Mapping



H−Slices

V−Columns



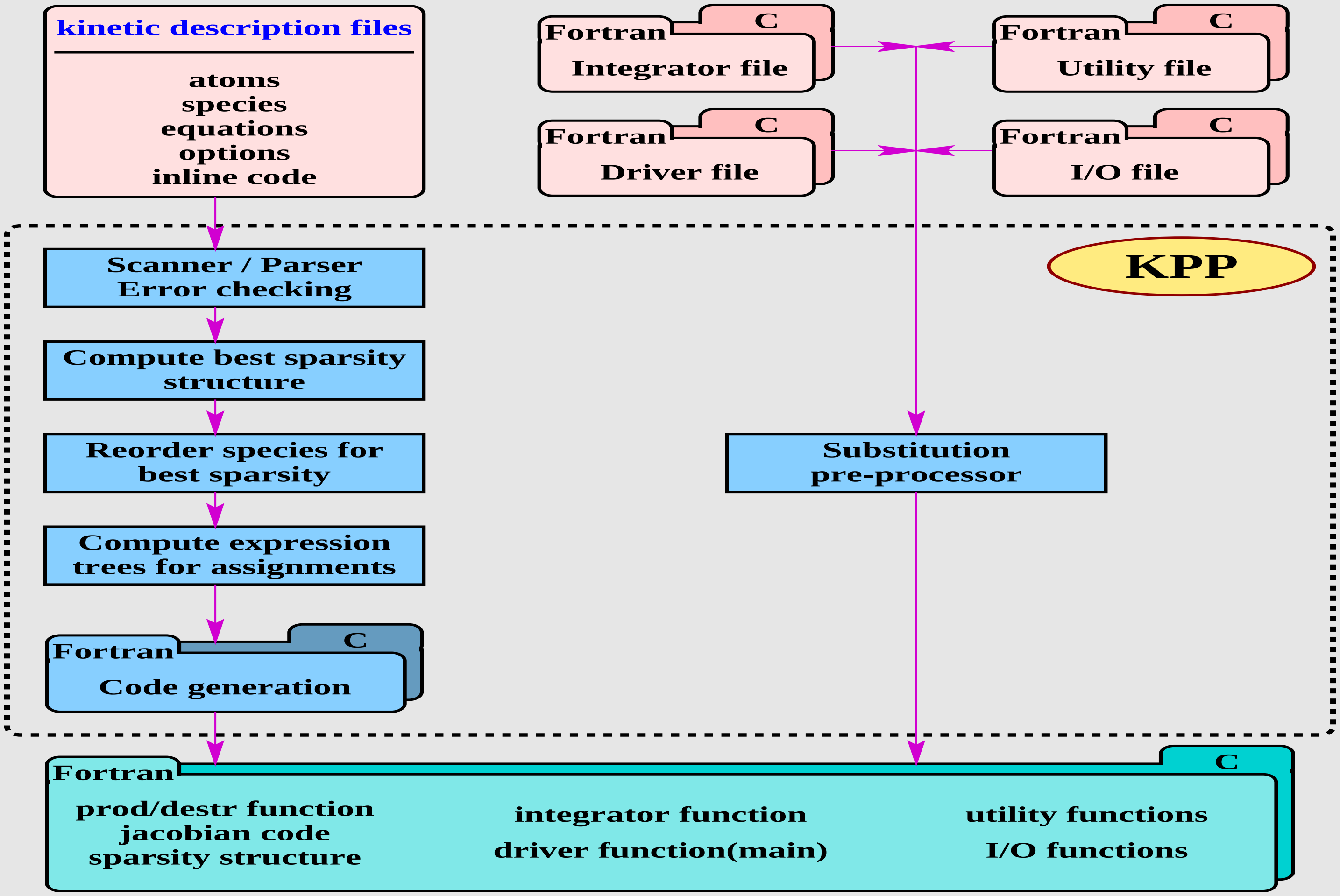
diagonal mapping

x/y−mapping by looking
at already assigned neighbors

diagonal mapping by looking
at already assigned neighbors

The Kinetic PreProcessor (KPP)

www.cs.mtu.edu/∼asandu/Software/KPP



kinetic description files

atoms
species
equations
options
inline code

Fortran C Integrator file

Fortran C Utility file

Fortran C Driver file

Fortran C I/O file

KPP

Scanner / Parser
Error checking

Compute best sparsity
structure

Reorder species for
best sparsity

Compute expression
trees for assignments

Fortran C Code generation

Substitution
pre-processor

Fortran C

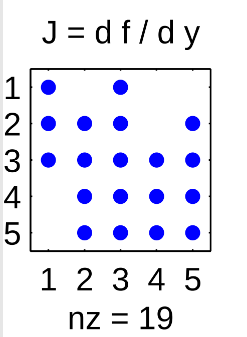
prod/destr function
jacobian code
sparsity structure

integrator function
driver function(main)

utility functions
I/O functions

KPP-1.2 Options for Adjoints

#JACOBIAN [ON | OFF | SPARSE]



JacVar(), JacVar_SP()

JacVar_SP_Vec(), JacVarTR_SP_Vec()

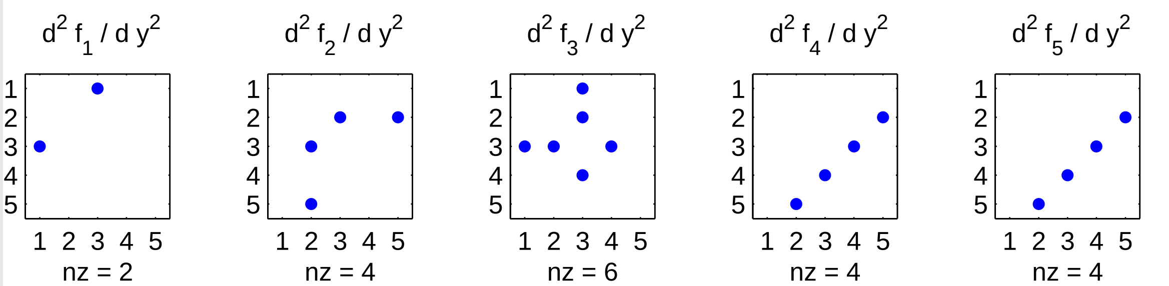
KppDecomp(), KppSolve(), KppSolveTR()

#HESSIAN [ON | OFF]

HessVar()

HessVar_Vec()

HessVarTR_Vec()



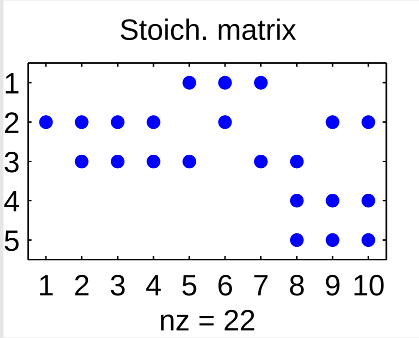
More KPP-1.2 Options for Adjoints

#STOICMAT [ON | OFF]

STOICM (col. compressed)

ReactantProd()

JacVarReactantProd()



Derivatives w.r.t. rate coefficients

dFunVar_dRcoeff(), dJacVar_dRcoeff()

Note: Code for direct and adjoint sensitivity calculations generated.

Algorithmic Developments

1. Discrete and continuous adjoints for stiff odes; formulation and theoretical properties.

2. Applications to stiff chemical mechanisms (box model): Data assimilation, adjoint sensitivity analysis, etc.

3. Discrete adjoints for aerosol dynamics (Sandu formulation); data assimilation for aerosol dynamics.

Adjoint STEM-III

1. Fully parallel adjoint of STEM-III complete.
2. SAPRC-99 chemical mechanism (can be easily changed).
3. Finite differences (3rd order upwind) transport, locally 1-D, fraction formulation.
4. 2-level checkpointing; level 1 parallel; level 2 in memory.
5. On 18 processors: 2 min./hour of forward integration plus another 2.5 min./hour of backwards (adjoint) integration. For assimilation on 24 hrs window, LBFGS 10 iterations, estimated cpu time = 18 hours.

To Do in Near Future

1. Validate STEM-III adjoint.
2. Choose state-of-science aerosol module and build adjoint. Use existing framework for solving coagulation and growth.
3. Add aerosol module to adjoint STEM-III.
4. Start assimilaion of real data fields.
5. Extend theoretical investigations of discrete adjoints for stiff ode's etc.