Tools for Chemical Data Assimilation into AQMs*

Adrian Sandu†

Dacian Daescu‡

Gregory R. Carmichael°

†Computer Science Department, Michigan Technological University

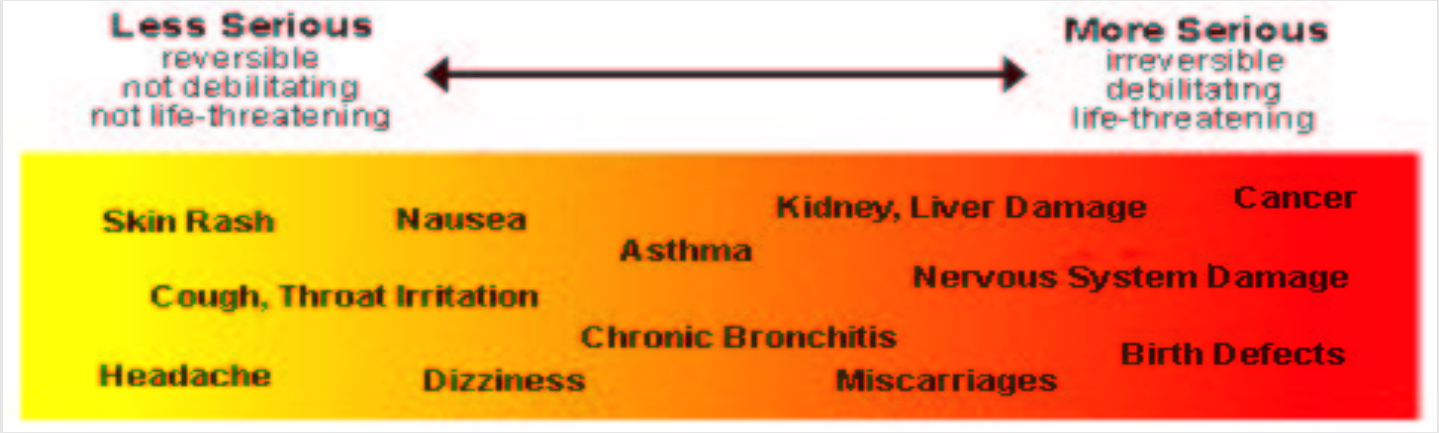
‡Institute for Mathematics and its Applications, University of Minnessotta

°CGRER, University of Iowa

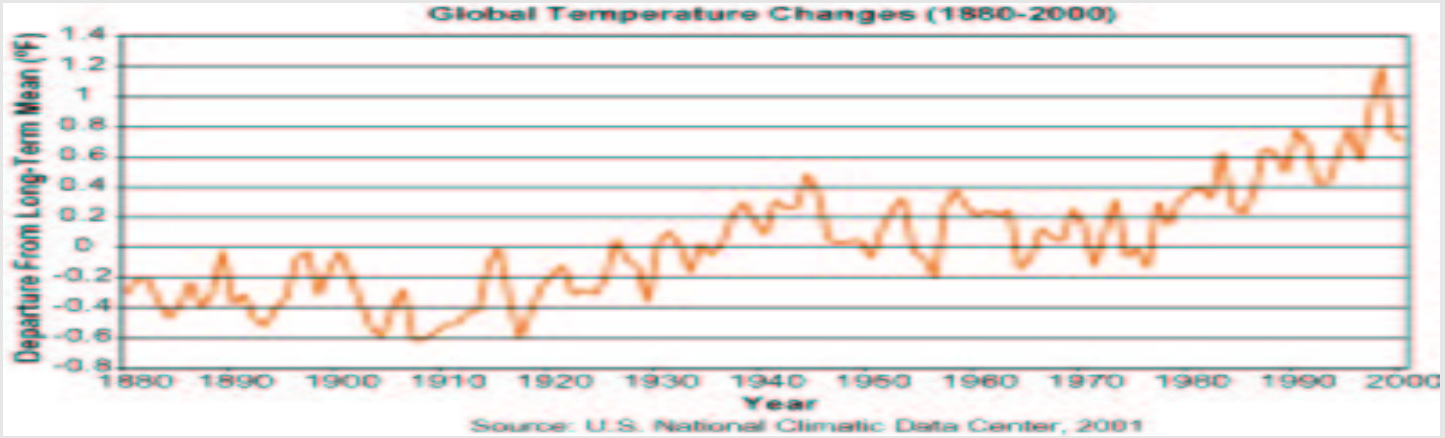
* NSF CAREER ACI-0093139 and NSF ITR AP &IM 0205198

Impacts of Air Pollution

(EPA, World Resources Institute)



$\rightarrow$ World Bank: $\sim$ 1% of global mortality related to air pollution.



$\rightarrow$ Expected 2-10° F temperature increase over 21st century.

Air Quality Measurements and Models

- Measurements: to asses air pollution levels.
- Models: to understand, predict, and regulate air pollution.
- AQMs simulate the fate of pollutants in the atmosphere

  $$\frac{\partial}{\partial t}C_i + \nabla(UC_i) = \nabla(K\nabla C_i) + \sum_{j=1}^{NP} R_i^j, \;\; i = 1, NS.$$

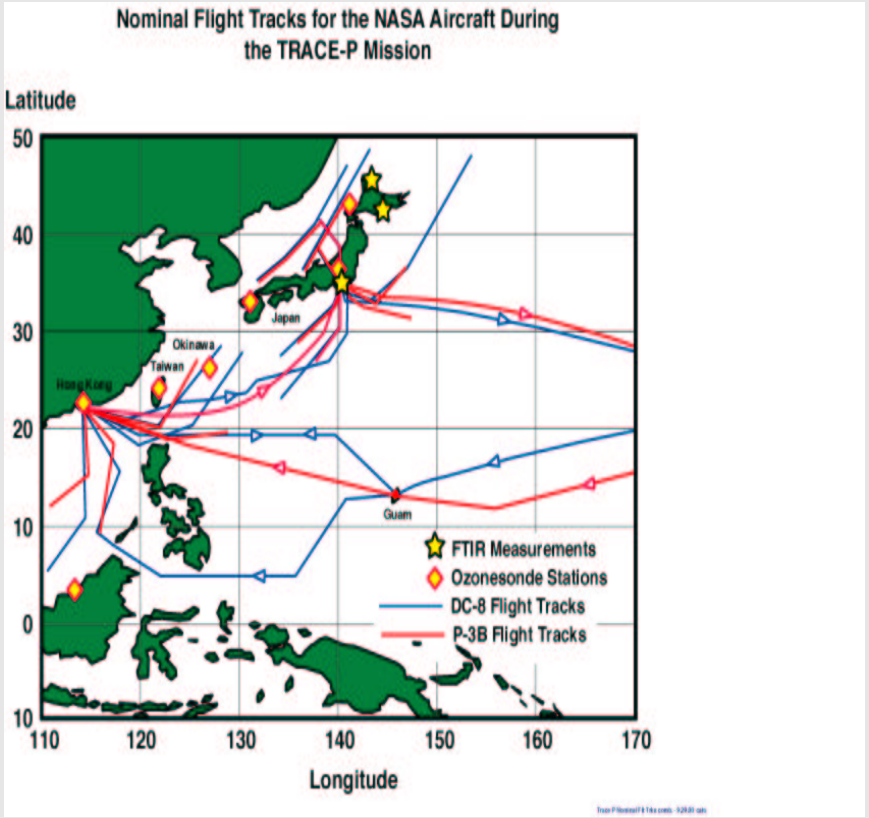
- Large, complex, computationally-intensive models;
- Operator Splitting (Strang)

  $$T_{\text{hor}}^{\Delta t} \cdot T_{\text{vert}}^{\Delta t} \cdot C^{2\Delta t} \cdot T_{\text{vert}}^{\Delta t} \cdot T_{\text{hor}}^{\Delta t} \;\Rightarrow\; \text{Split sensitivity model.}$$

Example: Trace-P Campaign

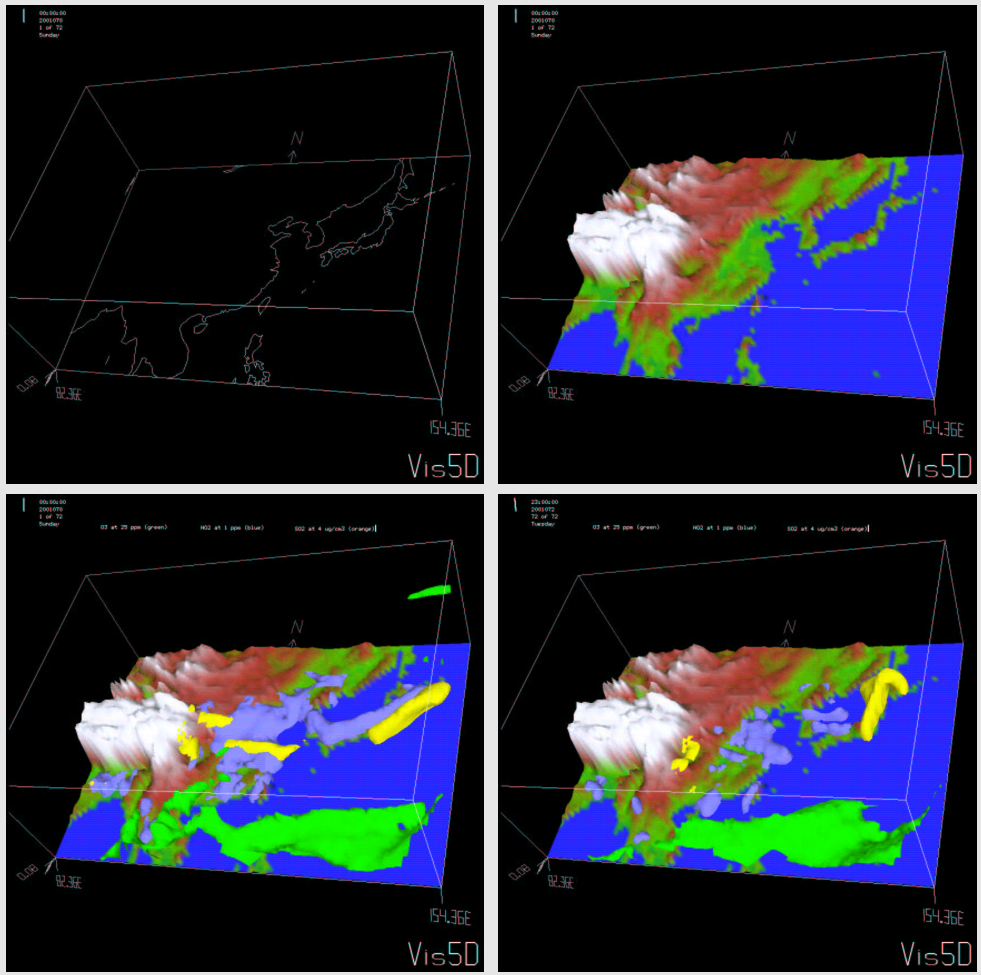
NASA Transport and Chemical Evolution over the Pacific (03–04/2001):

1) quantify Asian export; 2) understand chemistry over W. Pacific.



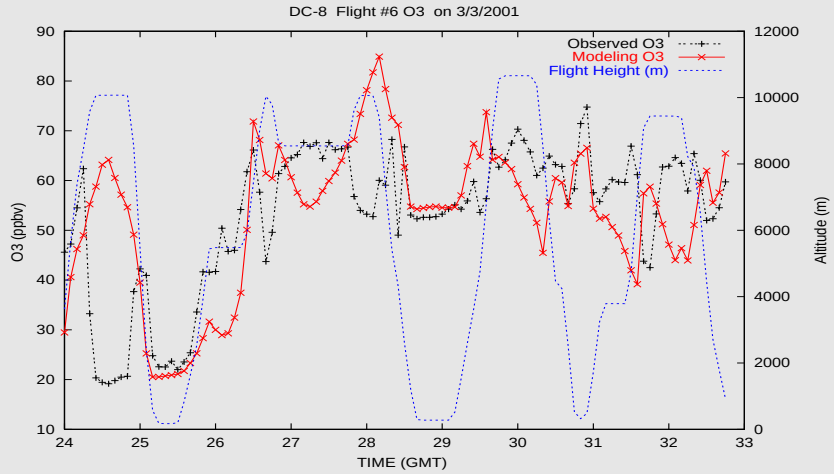
Example: East Asia Model

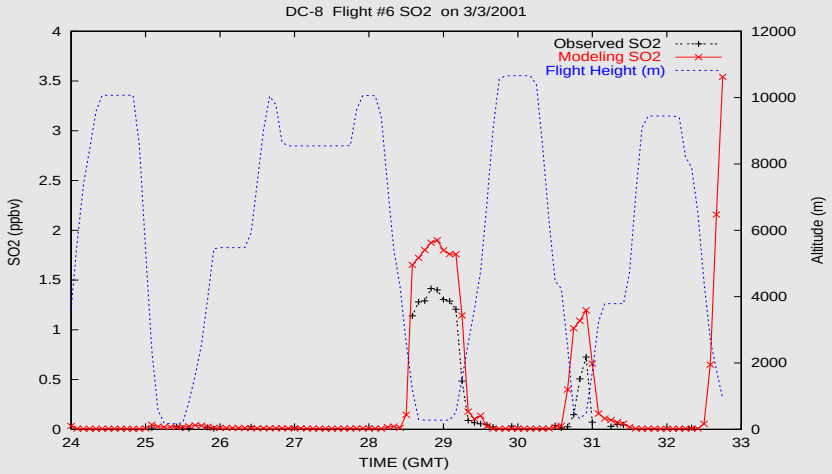
Simulated period: March 01-04 2001.

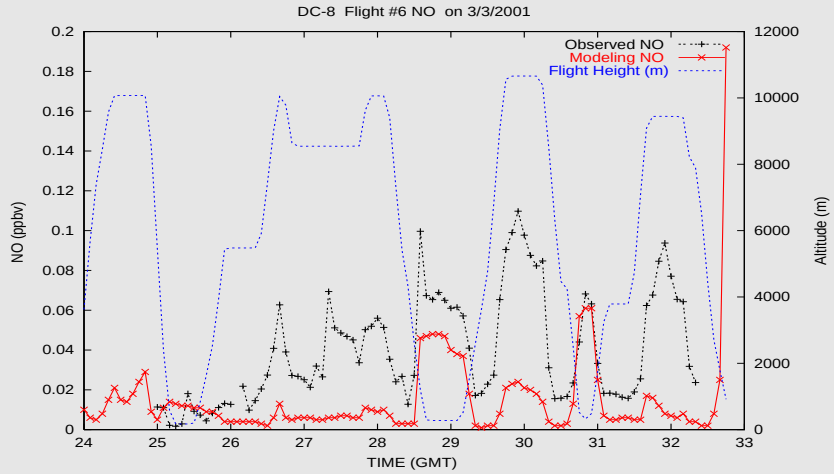


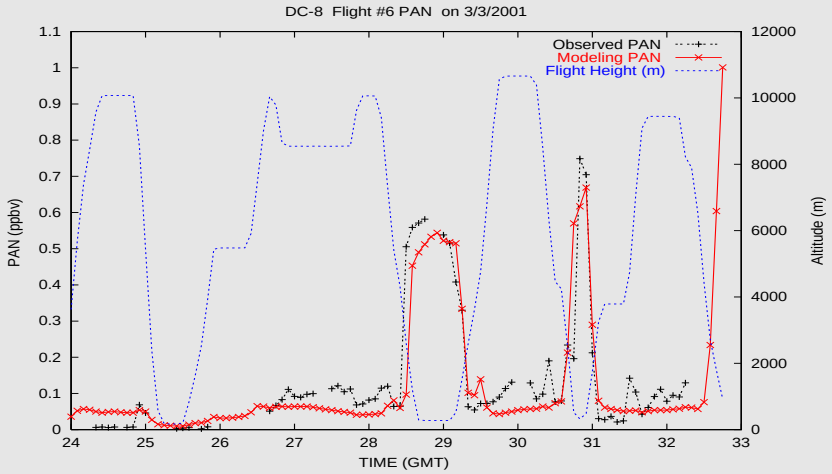
Trace-P: Model vs. Observations

(CGRER/U.Iowa)



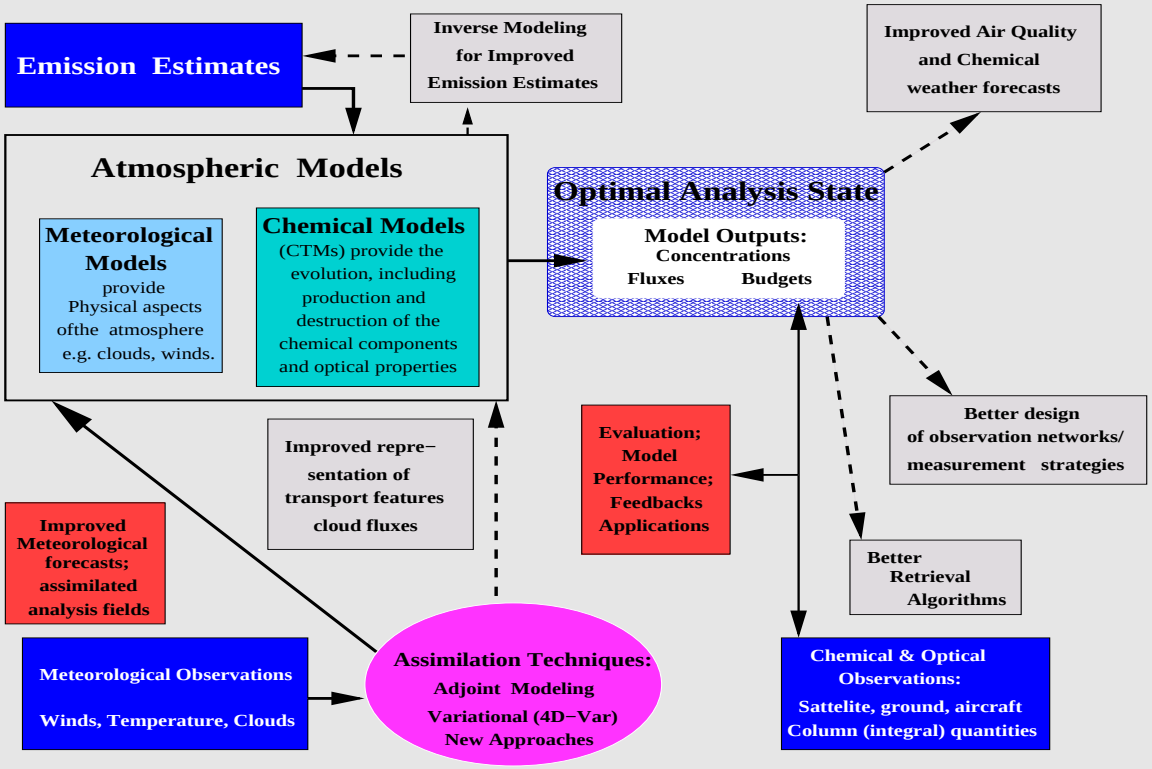






4D-Var Data Assimilation

Purpose: integrate info from chemical observations into model.



Method: $\min \mathcal{F}(y^0) = \frac{1}{2}\sum_k (H_k y^k - y^k_{\rm obs})^T \mathbf{R}_k^{-1} (H_k y^k - y^k_{\rm obs})$

Note: Optimizer needs $\mathcal{F}$ and $\nabla_{y^0}\mathcal{F}$ (adjoint).

Adjoint Sensitivity

Want: Stiff ODE (e.g. chemistry); scalar functional

$$y' = f(t,y)\,,\ t^0 \le t \le t^F\ ;\ \mathcal{F}(y) \quad \Rightarrow \quad \nabla_{y^0}\mathcal{F}(y(t^F))$$

Continous: Solve numerically

$$\begin{aligned} \lambda' &= -J^T(t,y)\lambda\,, \quad \lambda(t^{\mathrm{F}}) = \nabla_y \mathcal{F}|_{y(t^{\mathrm{F}})} \\ \nabla_{y^0}\mathcal{F} &= \lambda(t^0)\,. \quad \text{(store fwd traj. y(t))} \end{aligned}$$

Discrete: Derivative of numerical method ($\sim$ AD)

$$\begin{aligned} y^{i+1} &= F^i(y^i)\,, \quad i = 0,\ldots,N-1\,, \\ \lambda^i &= F^i_y(y^i,p^i)^T\lambda^{i+1}\,, \quad \lambda^N = \nabla_y \mathcal{F}\big|_{y^N} \\ \nabla_{y^0}\mathcal{F}(y^N) &= \lambda^0\,. \quad \text{(store fwd traj. y}^{\mathrm{i}}\text{)} \end{aligned}$$

Some Considerations

Stiffness: e.g. $\tau_O = 10^{-9}$ sec., $\tau_{O3} = 10^3$ sec., $\Delta t \sim 10^2$ sec.

Data assim. needs derivative of the model (discrete);

For stiff eqns. cts. adjoint (usually) easier to implement and faster;

Q: Relationship between continuous and discrete adjoints? (not easy: see e.g. Sei and Symes)

Hope: forward solution $\approx$ exact $\Rightarrow$ discrete adj. $\approx$ continuous adj.;

Note: the discrete adjoint could be viewed as a numerical discretization of the continuous adjoint eqn; this discretization is induced by the forward numerical method used.

Q: Is this induced discretization a “good” method to solve the continuous adjoint equation?

Linear Multistep Methods

The Method:

$$\sum_{i=0}^{k} \alpha_i^{[n]} y_{n-i} = h_n \sum_{i=0}^{k} \beta_i^{[n]} f(y_{n-i})$$

The Continuous Adjoint:

$$\sum_{i=0}^{k} \alpha_i^{[m]} \lambda_{m+i} = h_m \sum_{i=0}^{k} \beta_i^{[m]} J^T(y_{m+i}) \lambda_{m+i}$$

The Discrete Adjoint:

$$\sum_{i=0}^{k} a_i^{[m+i]} \lambda_{m+i} = J^T(y_m) \sum_{i=0}^{k} h_{m+i} \beta_i^{[m+i]} \lambda_{m+i}$$

$\sim$ one-leg method; in general not consistent with the continuous adjoint.

Runge-Kutta Methods

The Method:

$$y_{n+1} = y_n + h \sum_{j=1}^{s} b_j f(z_j) \,, \quad z_i = y_n + h \sum_{j=1}^{s} a_{ij} f(z_j) \,.$$

The Continuous Adjoint: $M_j = hJ^T\big(y(t_{n+1} - \tilde{c}_j h)\big)$

$$\lambda_n = \lambda_{n+1} + \sum_{j=1}^{s} \tilde{b}_j M_j \cdot w_j \,, \quad w_i = \lambda_{n+1} + \sum_{j=1}^{s} \tilde{a}_{ij} M_j \cdot w_j \,.$$

The Discrete Adjoint: (Hager) $L_i = hJ^T(z_i)$

$$\lambda_n = \lambda_{n+1} + \sum_{j=1}^{s} \theta_j \,, \quad \theta_i = L_i \left[b_i \lambda_{n+1} + \sum_{j=1}^{s} a_{ji} \theta_j \right]$$

Consistency of Runge Kutta Adjoints

(Sandu, Daescu, 2003)

Consistency: Consider a Runge-Kutta method of order p . Its discrete adjoint is an order p numerical discretization of the continuous adjoint equation. (Proof: use elementary differentials of transfer fcns).

Corollary: For explicit RK reverse automatic differentiation gives what we hope for.

Control Problems: (Hager) $f = f(y, \lambda)$. One can use the P-tree theory for partitioned differential equations to obtain the extra conditions of Hager, and generalize that theory.

Singular Perturbation Analysis

(Sandu, Daescu, 2003) Relevant for stiff problems.

$$y' = f(y,z)\ , \quad \epsilon z' = h(y,z)\ , \quad t^0 \le t \le t^F$$

Distinguish between derivatives w.r.t. non-/stiff variables:

$$\lambda = \partial \mathcal{F}\big(y(\cdot), z(\cdot)\big)/\partial y\ , \quad \mu = \partial \mathcal{F}\big(y(\cdot), z(\cdot)\big)/\partial z\ .$$

If: RK with invertible coefficient matrix A and $R(\infty) = 0$;

and the cost function depends only on the non-stiff variable y

Then: $\mu^0 = 0$ and λ are solved with the same accuracy as the original method, within $\mathcal{O}(\epsilon)$.

A similar conclusion holds for continuous Runge Kutta adjoints.

Rosenbrock Methods

(Sandu, Daescu, 2003)

The Method: $J_n = J(y_n)$

$$y_{n+1} = y_n + \sum_{j=1}^{s} m_j k_j \ , \quad Y_i = y_n + \sum_{j=1}^{i-1} a_{i,j} k_j \ ,$$
$$\left[\frac{1}{h\gamma} - J_n\right] k_i = f\left(Y_i\right) + \sum_{j=1}^{i-1} \frac{c_{i,j}}{h} k_j \ .$$

The Continuous Adjoint: $J_n = J(y_n)$

$$\lambda_n = \lambda_{n+1} + \sum_{j=1}^{s} m_j k_j \ , \quad \Lambda_i = \lambda_{n+1} + \sum_{j=1}^{i-1} a_{i,j} k_j \ ,$$
$$\left[\frac{1}{h\gamma} - J_{n+1}^T\right] k_i = J^T\Big(y(t_{n+1} - \alpha_i h)\Big) \cdot \Lambda_i + \sum_{j=1}^{i-1} \frac{c_{i,j}}{h} k_j \ ,$$

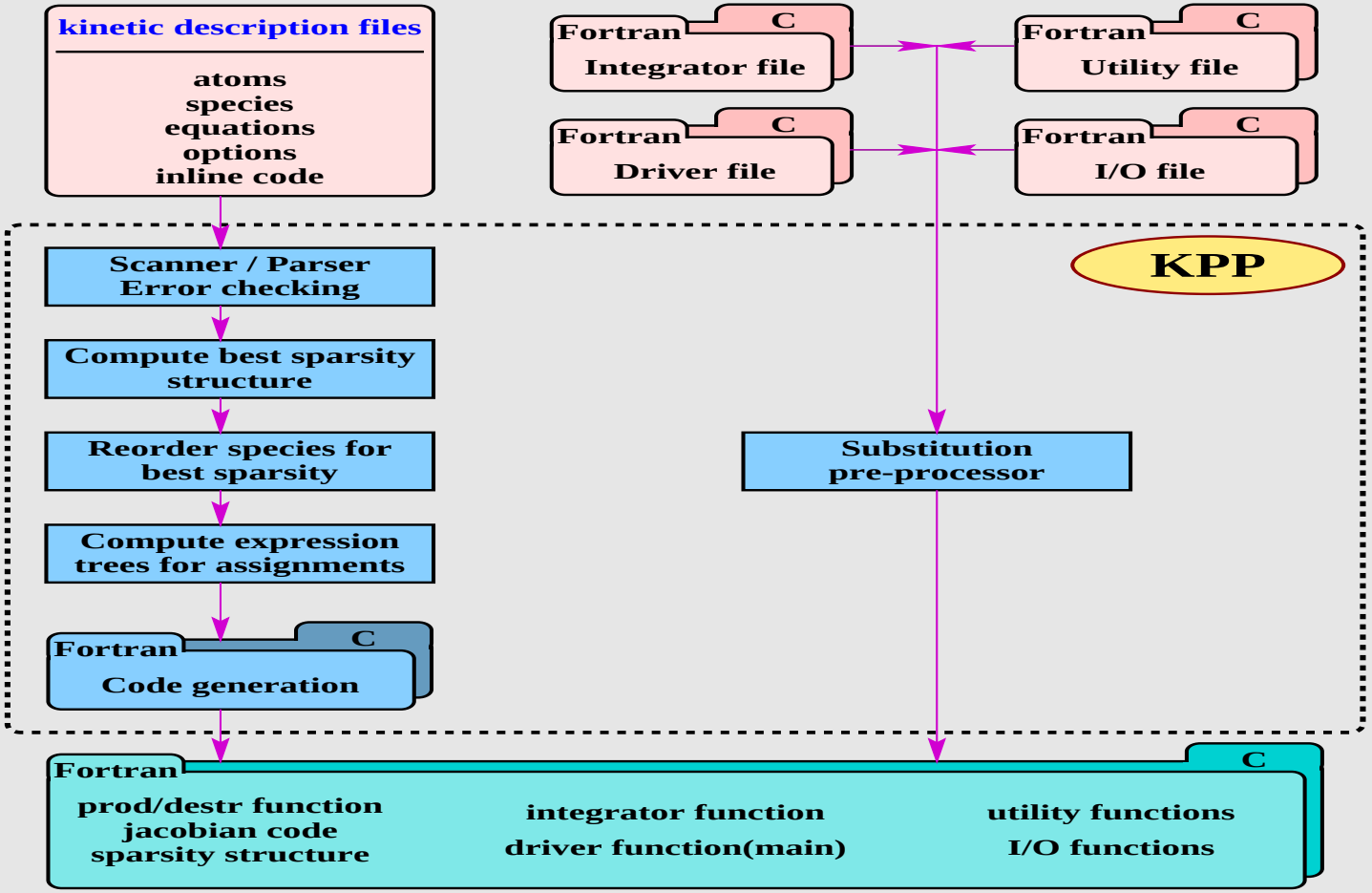
The Discrete Adjoint: $J_n = J(y_n), H_n = H(y_n)$

$$\lambda_n = \lambda_{n+1} + \sum_{i=1}^{s} \left(H_n \times k_i\right)^T \cdot u_i + \sum_{i=1}^{s} v_i \ , \quad v_i = J^T\left(Y_i\right) \cdot u_i$$
$$\left[\frac{1}{h\gamma} - J_n^T\right] u_i = m_i \lambda_{n+1} + \sum_{j=i+1}^{s} \left(a_{j,i} v_j + \frac{c_{j,i}}{h} u_j\right) \ .$$

The Kinetic PreProcessor (KPP)

(Damian, Sandu, et. al. 1996); (Sandu et. al. 2001, 2002)

http://www.cs.mtu.edu/∼asandu/Software/Kpp



Example, Model Description

```
#INCLUDE atoms

#DEFVAR

O       =    O;
O1D     =    O;
O3      =    O + O + O;
NO      =    N + O;
NO2     =    N + O + O;

#DEFFIX

O2      =    O + O;
M       =    ignore;
```

```
#EQUATIONS { Small Stratospheric }

O2 + hv = 2O              :    2.6E-10*SUN**3;
O + O2 = O3               :    8.0E-17;
O3 + hv = O + O2          :    6.1E-04*SUN;
O + O3 = 2O2              :    1.5E-15;
O3 + hv = O1D + O2        :    1.0E-03*SUN**2;
O1D + M = O + M           :    7.1E-11;
O1D + O3 = 2O2            :    1.2E-10;
NO + O3 = NO2 + O2        :    6.0E-15;
NO2 + O = NO + O2         :    1.0E-11;
NO2 + hv = NO + O         :    1.2E-02*SUN;
```

Example, Fortran Code

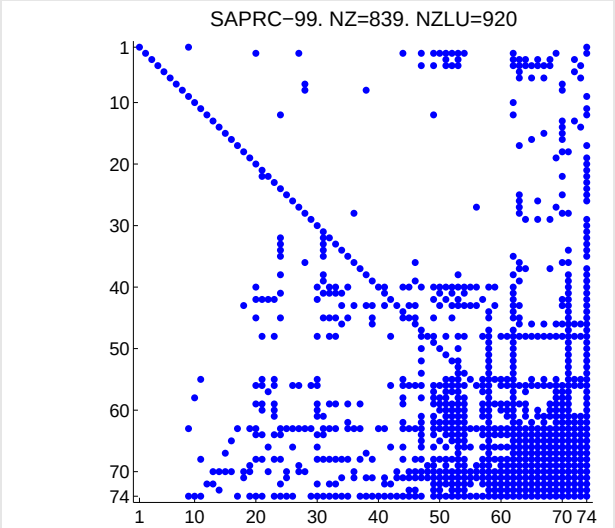
```
      SUBROUTINE FunVar ( V, R, F, RCT, A_VAR )
      INCLUDE 'small.h'
      REAL*8 V(NVAR), R(NRAD), F(NFIX), RCT(NREACT), A_VAR(NVAR)
C A - rate for each equation
      REAL*8 A(NREACT)
C Computation of equation rates
      A(1) = RCT(1)*F(2)
      A(2) = RCT(2)*V(2)*F(2)
      A(3) = RCT(3)*V(3)
      A(4) = RCT(4)*V(2)*V(3)
      A(5) = RCT(5)*V(3)
      A(6) = RCT(6)*V(1)*F(1)
      A(7) = RCT(7)*V(1)*V(3)
      A(8) = RCT(8)*V(3)*V(4)
      A(9) = RCT(9)*V(2)*V(5)
      A(10) = RCT(10)*V(5)
C Aggregate function
      A_VAR(1) = A(5)-A(6)-A(7)
      A_VAR(2) = 2*A(1)-A(2)+A(3)-A(4)+A(6)-A(9)+A(10)
      A_VAR(3) = A(2)-A(3)-A(4)-A(5)-A(7)-A(8)
      A_VAR(4) = -A(8)+A(9)+A(10)
      A_VAR(5) = A(8)-A(9)-A(10)
      RETURN
      END
```

KPP / Sparse Jacobians

#JACOBIAN [ON | OFF | SPARSE]

- JacVar(), JacVar_SP()
- JacVar_SP_Vec(), JacVarTR_SP_Vec()
- KppDecomp(), KppSolve(), KppSolveTR()

E.g. SAPRC-99 (74+5 spc. / 211 r.)



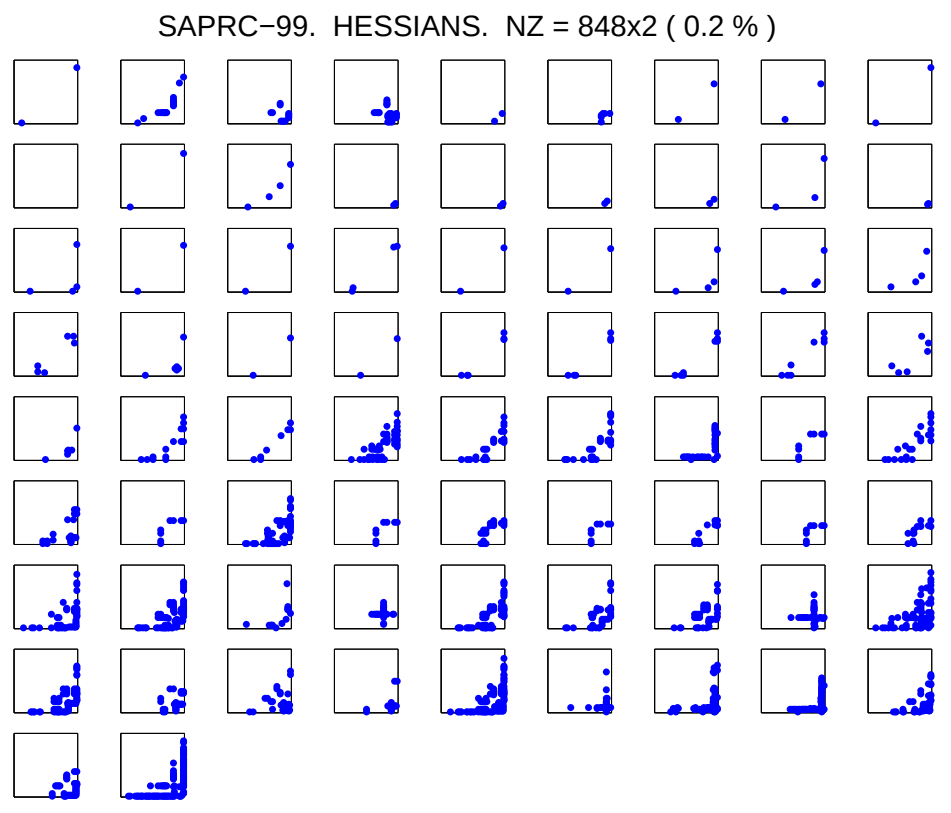
- Chem. interactions $\Rightarrow$ sparsity of $I - h\gamma J$ (off-line);
- Doolittle (non-pivoting) LU + loopfree substitution;

Harwell/Lapack_o	0.61	0.21	0.35
Kpp/Lapack_o	0.23	0.06	0.12

KPP / Sparse Hessians

#HESSIAN [ON | OFF]

- 3-tensor in sparse coordinate format; symmetry considered;
- HessVar(), HessVar_Vec(), HessVarTR_Vec()



KPP / Stoichiometric Form

#STOICMAT [ON | OFF]

STOICM (col. compressed)

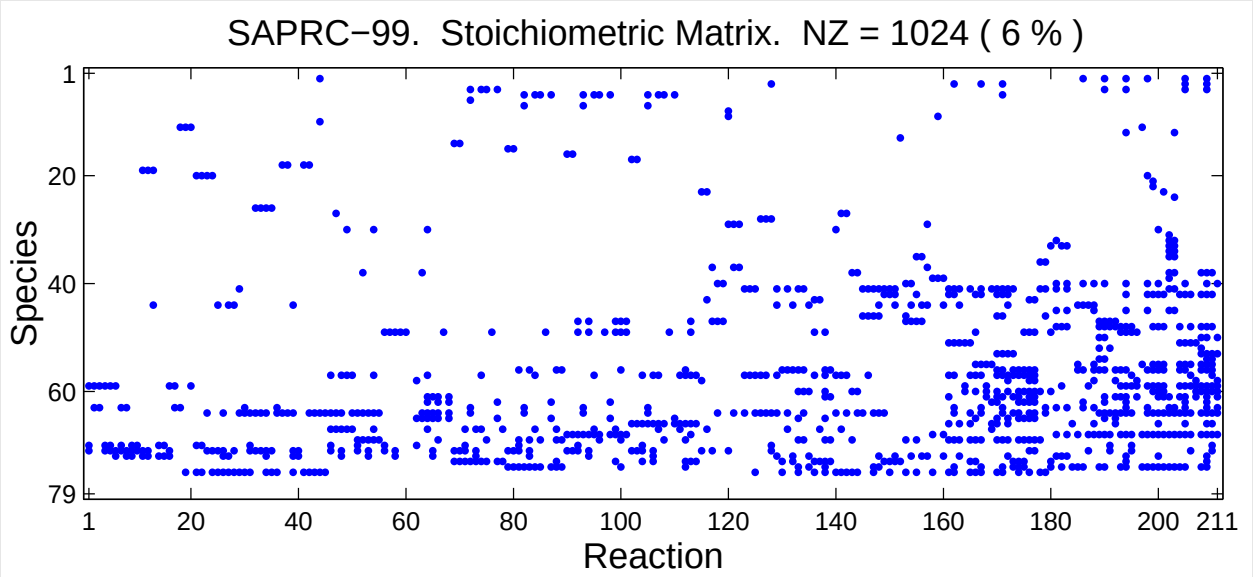
ReactantProd()

JacVarReactantProd()

Rcoeff Derivatives:

dFunVar_dRcoeff()

dJacVar_dRcoeff()



Direct-Decoupled Sensitivity w/ KPP

BDF (Dunker,1984): $S_\ell(t) = \partial y(t)/\partial p_\ell$

$$
\begin{aligned}
y^{n+1} &= \sum_{i=0}^{k-1} \alpha_i y^{n-i} + h\beta f^{n+1}\,, \\
[I - h\beta J_{n+1}]\; S_\ell^{n+1} &= \sum_{i=0}^{k-1} \alpha_i S_\ell^{n-i} + f_{\alpha_\ell}^{n+1} \quad (1 \le \ell \le m)\,.
\end{aligned}
$$

Rosenbrock (Sandu, Daescu, Carmichael, 2002):

$$
y^{n+1} = y^n + \sum_{i=1}^{s} m_i k_i^0, \quad S_\ell^{n+1} = S_\ell^n + \sum_{i=1}^{s} m_i k_i^\ell
$$

$$
T^i = t^n + \alpha_i h\,, \quad Y^i = y^n + \sum_{j=1}^{i-1} a_{ij} k_j^0\,,
$$

$$
\begin{aligned}
\left[\tfrac{1}{h\gamma} - J_n\right] k_i^0 &= f\left(T^i, Y^i; p\right) + \sum_{j=1}^{i-1} \tfrac{c_{ij}}{h} k_j^0 + h\gamma_i f_t^n \\
\left[\tfrac{1}{h\gamma} - J_n\right] k_i^\ell &= J\left(T^i, Y^i; p\right) \bullet \left(S_\ell^n + \sum_{j=1}^{i-1} a_{ij} k_j^\ell\right) \\
&\quad + f_{\alpha_\ell}\left(T^i, Y^i; p\right) + \sum_{j=1}^{i-1} \tfrac{c_{ij}}{h} k_j^\ell + J_{\alpha_\ell}^n \bullet k_i^0 \\
(1 \le \ell \le m) &\quad + \left(H_n \times S_\ell^n\right) k_i^0 + h\gamma_i J_t^n S_\ell^n + h\gamma_i f_{\alpha_\ell, t}^n
\end{aligned}
$$

Continuous Adjoint Sensitivity w/ KPP

(Sandu, Daescu, Carmichael, 2002)

Runge-Kutta:

$$\begin{aligned} \lambda_n &= \lambda_{n+1} + h\sum_{j=1}^{s} \tilde{b}_j J^T(t_{n+1} - c_j h) \bullet w_j \ , \\ w_i &= \lambda_{n+1} + h\sum_{j=1}^{s} \tilde{a}_{ij} J^T(t_{n+1} - c_j h) \bullet w_j \ . \end{aligned}$$

Rosenbrock:

$$\begin{aligned} \lambda_n &= \lambda_{n+1} + \sum_{j=1}^{s} m_j k_j \ , \\ \Lambda_i &= \lambda_{n+1} + \sum_{j=1}^{i-1} a_{i,j} k_j \ , \\ \left[\frac{1}{h\gamma} - J^T\left(t_{n+1}\right)\right] k_i &= J^T\left(t_{n+1} - \alpha_i h\right) \bullet \Lambda_i + \sum_{j=1}^{i-1} \frac{c_{i,j}}{h} k_j \ . \end{aligned}$$

Discrete Adjoint Sensitivity w/ KPP

(Daescu, Sandu, Carmichael, 2002)

Runge-Kutta: (Hager)

$$\begin{aligned} \lambda_n &= \lambda_{n+1} + \textstyle\sum_{j=1}^{s} \theta_j \ , \\ \theta_i &= h\, J^T(z_i) \bullet \left[b_i \lambda_{n+1} + \textstyle\sum_{j=1}^{s} a_{ji} \theta_j \right] \end{aligned}$$

Rosenbrock: $J_n = J(y_n), H_n = H(y_n)$

$$\begin{aligned} \lambda_n &= \lambda_{n+1} + \textstyle\sum_{i=1}^{s} \left(H_n \times k_i\right)^T \cdot u_i + \sum_{i=1}^{s} v_i \ , \\ \left[\tfrac{1}{h\gamma} - J_n^T\right] u_i &= m_i \lambda_{n+1} + \textstyle\sum_{j=i+1}^{s} \left(a_{j,i} v_j + \tfrac{c_{j,i}}{h} u_j \right) \ , \\ v_i &= J^T(Y_i) \bullet u_i \ . \end{aligned}$$

SAPRC Data Assimilation

(Daescu, Sandu, Carmichael, 2002)

- SAPRC (Carter, 1999): 79 spec., 211 react. Emissions considered.
- Assim. window = first 24 hrs (of 120 hrs run). Hourly observations.

$$\mathcal{F}(\ln c^0, \ln E) = \frac{1}{2} \sum_{t=1}^{24\ h} \sum_{i \in Obs} \left[\ln c_i^k - \ln (c^{\mathrm{obs}})_i^k \right]^2$$

- Twin experiments (Em.+50%). LBFGS to decrease $\mathcal{F}$ 100 times.
- $T_{\mathrm{Adj.Grad}}$ (KPP/Rodas-3) $\left\{$ Griewank: $T_{\mathrm{dadj}} \leq 5 \times T_{\mathrm{fwd}}$ $\right\}$

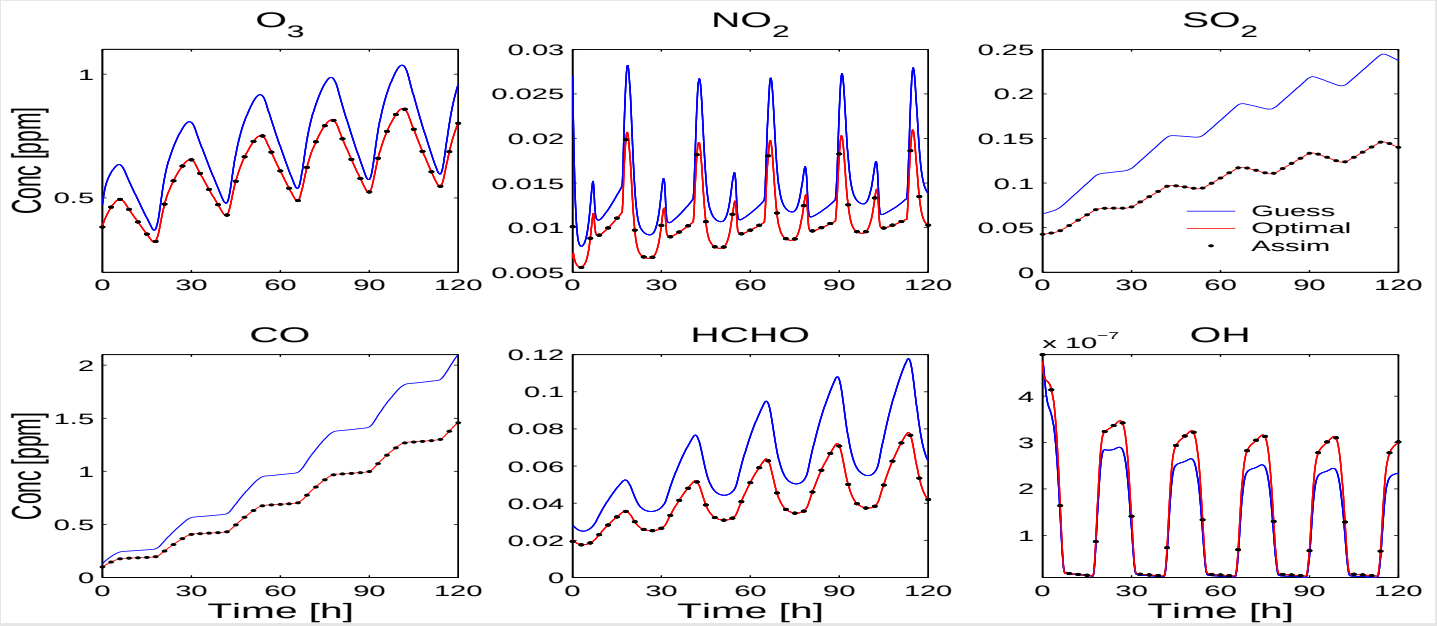
$T_{\mathrm{cadj}}/T_{\mathrm{fwd}}$	$T_{\mathrm{cadj}}^{\nabla\mathcal{F}}/T_{\mathrm{fwd}}$	$T_{\mathrm{dadj}}/T_{\mathrm{fwd}}$	$T_{\mathrm{dadj}}^{\nabla\mathcal{F}}/T_{\mathrm{fwd}}$
1.2	3.31	2.3	4.43

SAPRC Assimilation, cont'd

Assimilation of Emission Rates

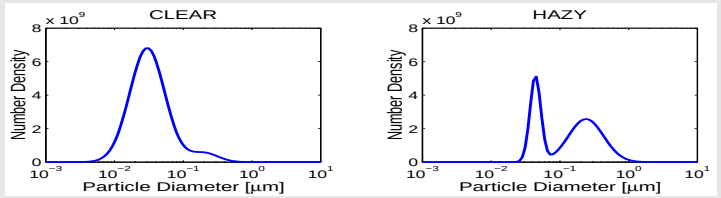
Species	NO	NO_2	SO_2	HCHO	ALK1
$\Delta E^0/E$	50%	50%	50%	50%	50%
$\Delta E^A/E$	0.13%	0.07%	0.26%	1.05%	0.94%

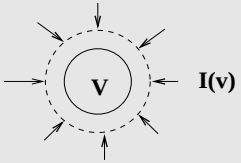
Assimilation of Initial Values



Aerosol Dynamics

n(v,t) = n. dens. [part/cm^3/μm^3].



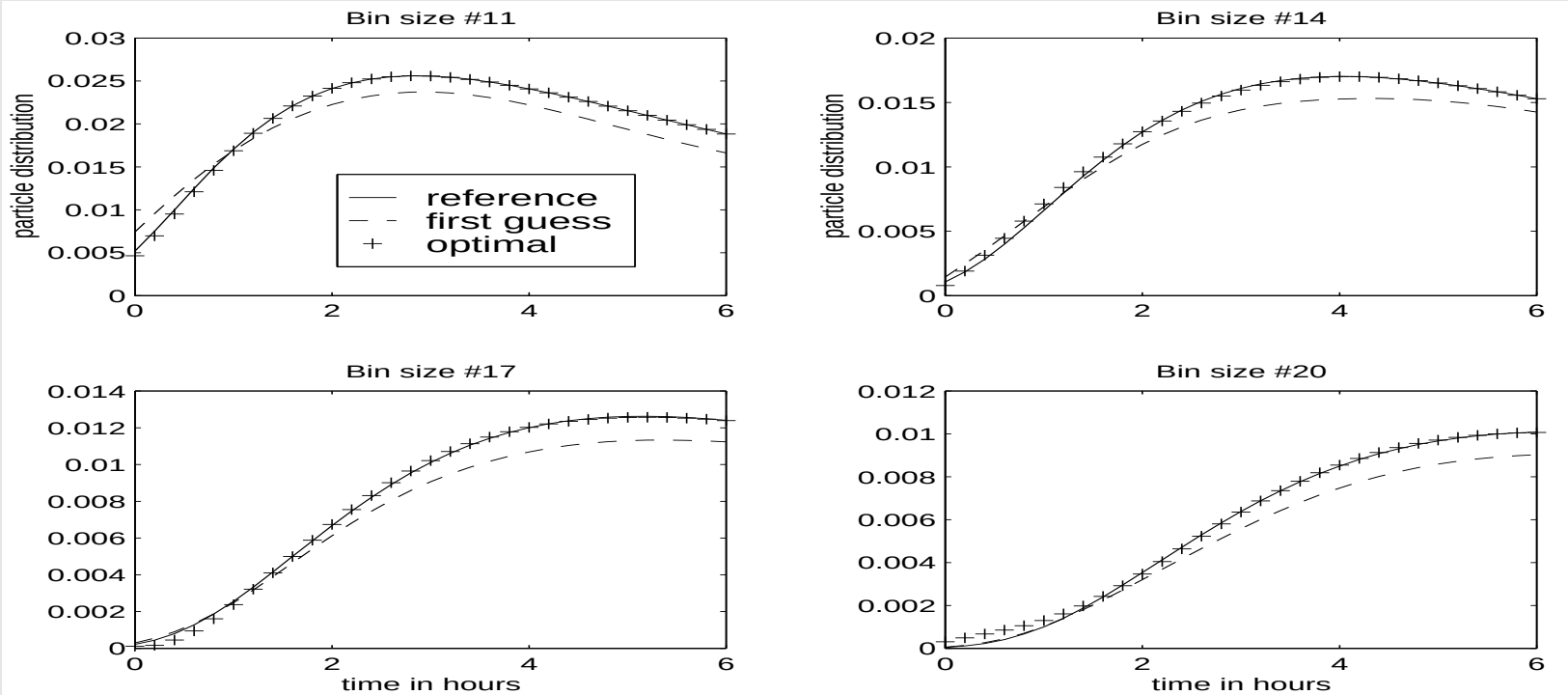


$$
\begin{aligned}
&\partial n(v,t)/\partial t = \\
&-\partial \left[I_v(v)\, n(v,t) \right] / \partial v && \text{(growth)} \\
&+\tfrac{1}{2} \int_0^v \beta_{v-w,w}\, n(v-w,t) n(w,t) dw && \text{(coag.+)} \\
&-n(v,t) \int_0^\infty \beta_{v,w}\, n(w,t) dw && \text{(coag.$-$)} \\
&+S(v,t) - R(v,t)\, n(v,t)\,, && \text{(src., dep.)} \\
&n(v,0) = n_0(v)\,, \quad n(0,t) = 0\,. && \text{(init, bdry).}
\end{aligned}
$$

Aerosol Dynamics Data Assimilation

(Sandu, Daescu, Carmichael, 2002)

Observations in only few size bins are provided. Data assimilation is able to retrieve the particle size distribution of unobserved bins using only information from the observed bins.



Parallelization

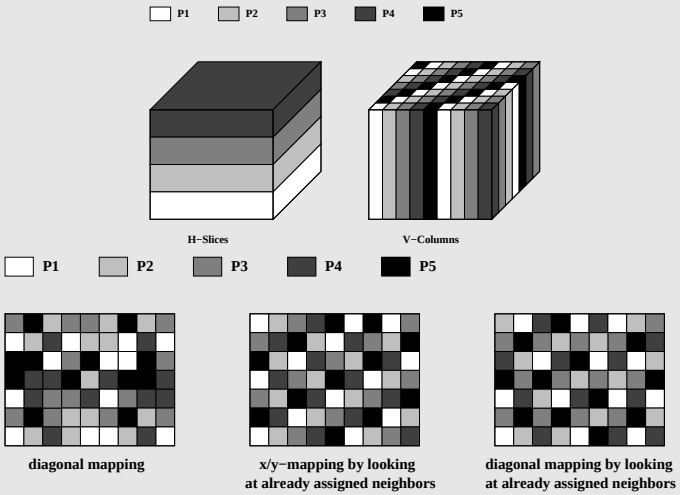
(Miehe and Sandu, 2001). http://www.cs.mtu.edu/∼asandu/Parallel

PAQMSG: Communication Library (F90/MPI)

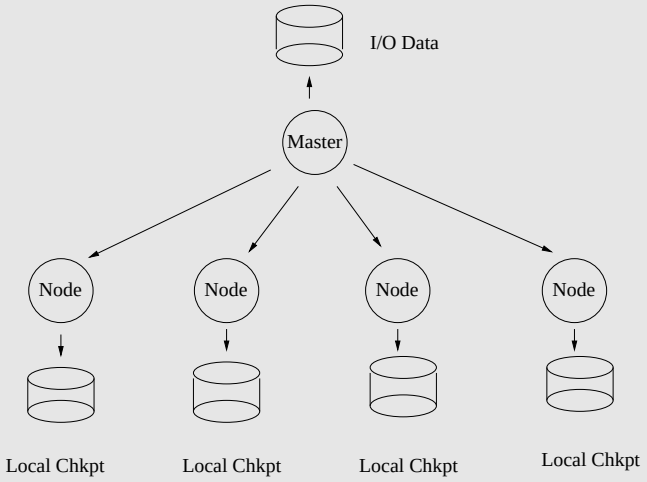
– Master/worker, domain decomp., static partitioning;

– 4D/3D/2D arrays; Alloc (local/global); Distrib; Gather; Shuffle.

Example Dom. Dec. & Mapping



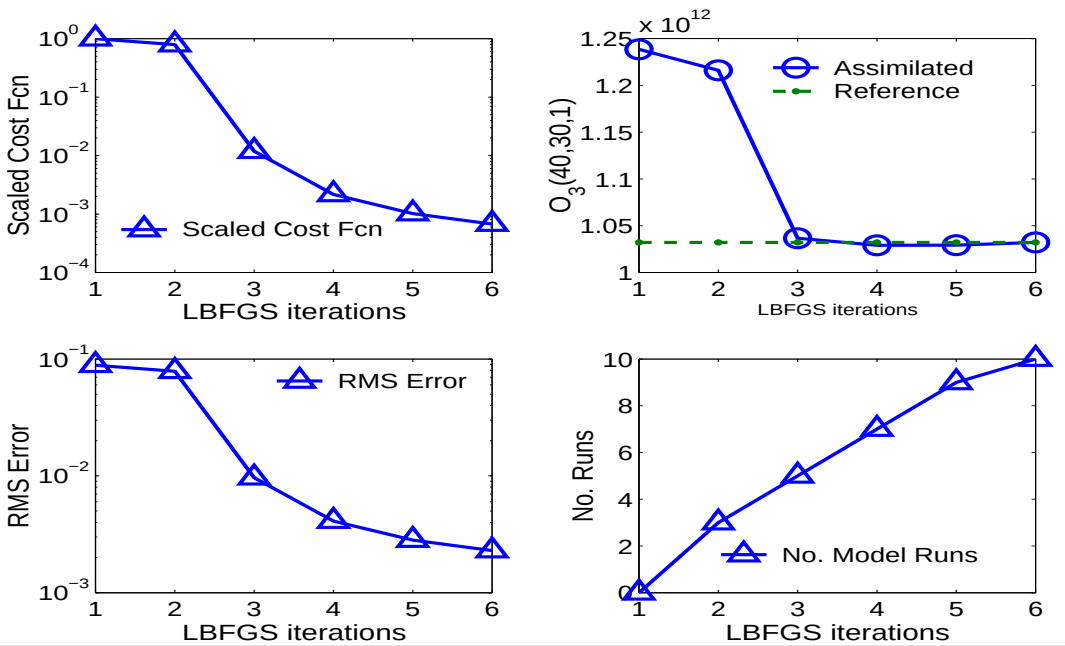
Parallel Checkpointing



3D Assimilation Results

- STEM III; ○ SAPRC 99 (Ros-2); ○ 3rd upwind FD, Crank-Nich;
- TraceP: [0,6] GMT March 01, 2001;
- Twin Experiments (O_3 +20%).

$$\mathcal{F} = \frac{1}{2} \sum_{\text{gridpoints}} [O_3(t^F) - O_3^{\text{obs}}(t^F)]^2$$



Conclusions and Future Work

Conclusions:

– Nonlinear, stiff chemistry important part of chemical transport models;

– One-step methods (RK, Ros) seem preferable w/ adjoint sensitivity;

– KPP: efficient code for simulation and direct/adjoint sensit.;

– Preliminary 3D asimilation results with parallel adjoint STEM-III.

Future work:

– Analysis of Rosenbrock;

– Assimilate real measurement data (TraceP, AceAsia);

– Complex transport schemes (FV, DG);

– Optimal placement of measurements (space, time, species);

– Optimal emission control strategies.