KPP – A Software Environment for the Simulation of Chemical Kinetics

Adrian Sandu
Computer Science Department
Virginia Tech

NIST, May 25, 2004

Acknowledgements

- Dr. Valeriu Damian
- Dr. Florian Potra
- Dr. Gregory Carmichael
- Dr. Dacian Daescu
- Dr. Tianfeng Chai
- Dr. Mirela Damian

Part of this work is supported by

- NSF ITR AP&IM-0205198

Related Work

CHEMKIN (http://www.reactiondesign.com/lobby/open)

Translates symbolic chemical system in a data file that is then used by internal libraries for simulation. Gas-phase kinetics, surface kinetics, reversible equations, transport, mixing, deposition for different types of reactors, direct sensitivity analysis (Senkin). Database of reaction data, graphical postprocessor for results.

KINALC (http://www.chem.leeds.ac.uk/Combustion/kinalc.htm)

Postprocessor to CHEMKIN for sensitivity and uncertainty analysis, parameter estimation, and mechanism reduction; etc.

KINTECUS (http://www.kintecus.com)

Compiler/ chemical modeling software. Can run heterogeneous and equilibrium chemistry, generates analytical Jacobians, fit/optimize rate constants, initial concentrations, etc. from data; sensitivity analysis; Excel interface. Can use Chemkin models and databases.

CANTERA (http://rayleigh.cds.caltech.edu/~goodwin/cantera)

Object-oriented package for chemically-reacting flows. C++ kernel, STL, standard numerical libraries, Interfaces for MATLAB, Python, C++, and Fortran. Capabilities: homogeneous and heterogeneous kinetics, equilibria, reactor modeling, multicomponent transport.

LARKIN/LIMKIN (http://www.zib.de/nowak/codes/limkin_1.0/full)

Simulation of LARge systems of chemical reaktion KINetics and parameter identification. Parser generates Fortran code for function and Jacobian, or internal data arrays.

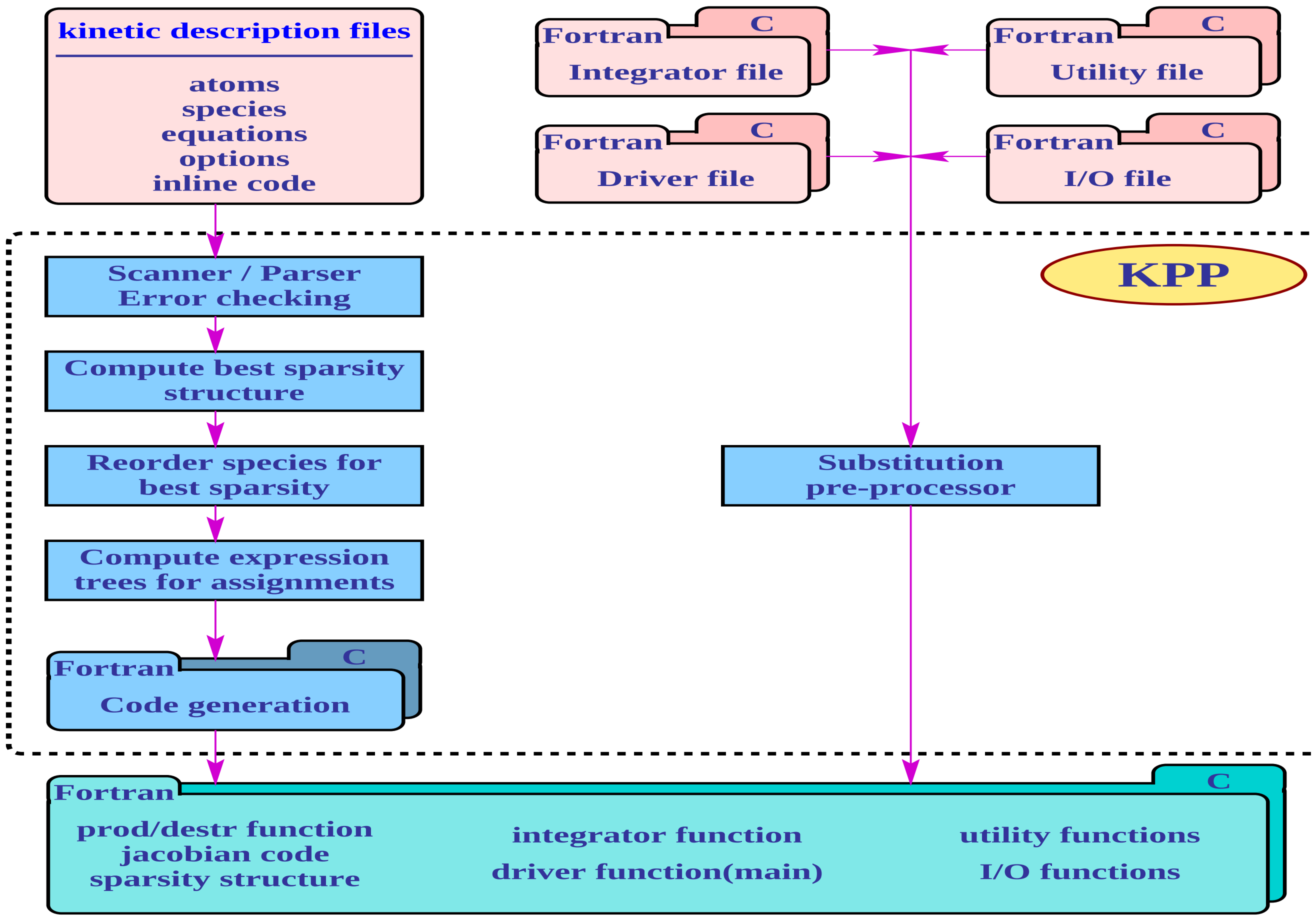
DYNAFIT (http://www.biokin.com/dynafit)

Performs nonlinear least-squares regression of chemical kinetic, enzyme kinetic, or ligand-receptor binding data using experimental data. Parses symbolic equations.

KPP in a Nutshell

- The Kinetic PreProcessor
- *Purpose: automatically implement building blocks for large-scale simulations*
- Parses chemical mechanisms
- Generates Fortran and C code for simulation, and direct and adjoint sensitivity analysis
- Function, Jacobian, Hessian, Stoichiometric matrix, derivatives of function&Jacobian w.r.t. rate coefficients
- Treatment of sparsity
- Comprehensive library of numerical integrators
- Used in several countries by academia/research/industry
- Free! http://www.cs.vt.edu/~asandu/Software/KPP

KPP Architecture



KPP Example

```
#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;
```

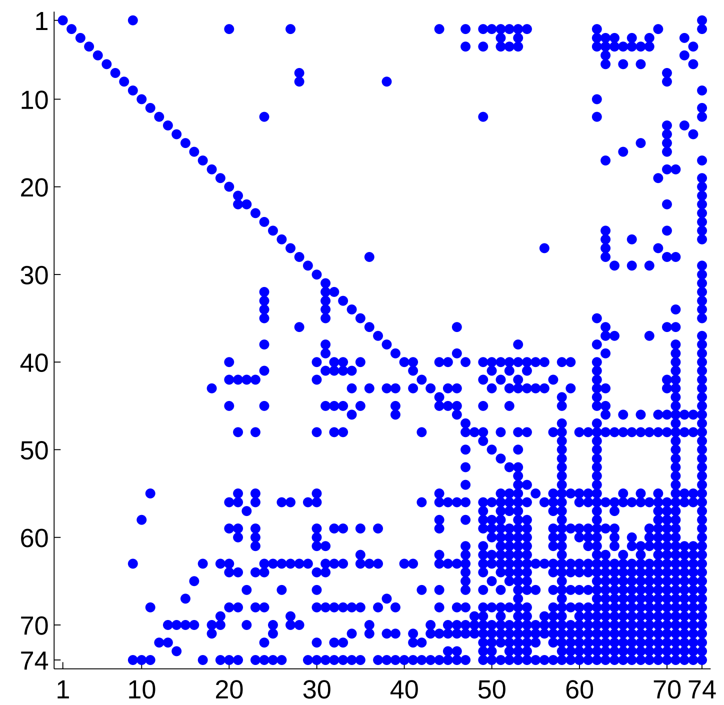
```
      SUBROUTINE FunVar ( V, R, F, RCT, A_VAR )
      INCLUDE 'small.h'
      REAL*8 V(NVAR), R(NRAD), F(NFIX)
      REAL*8 RCT(NREACT), A_VAR(NVAR)
C A - rate for each equation
      REAL*8 A(NREACT)
C Computation of equation rates
      A(1) = RCT(1)*F(2)
      A(2) = RCT(2)*V(2)*F(2)
      A(3) = 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
```

Sparse Jacobians

IDEAS:

- Chem. interactions: sparsity pattern (off-line)
- Min. fill-in reordering
- Expand sparsity structure
- Row compressed form
- Doolitle LU factorization
- Loop-free substitution

E.g. SAPRC-99
74+5 spc./211 react.
NZ=839, NZLU=920



#JACOBIAN [ON | OFF | SPARSE]

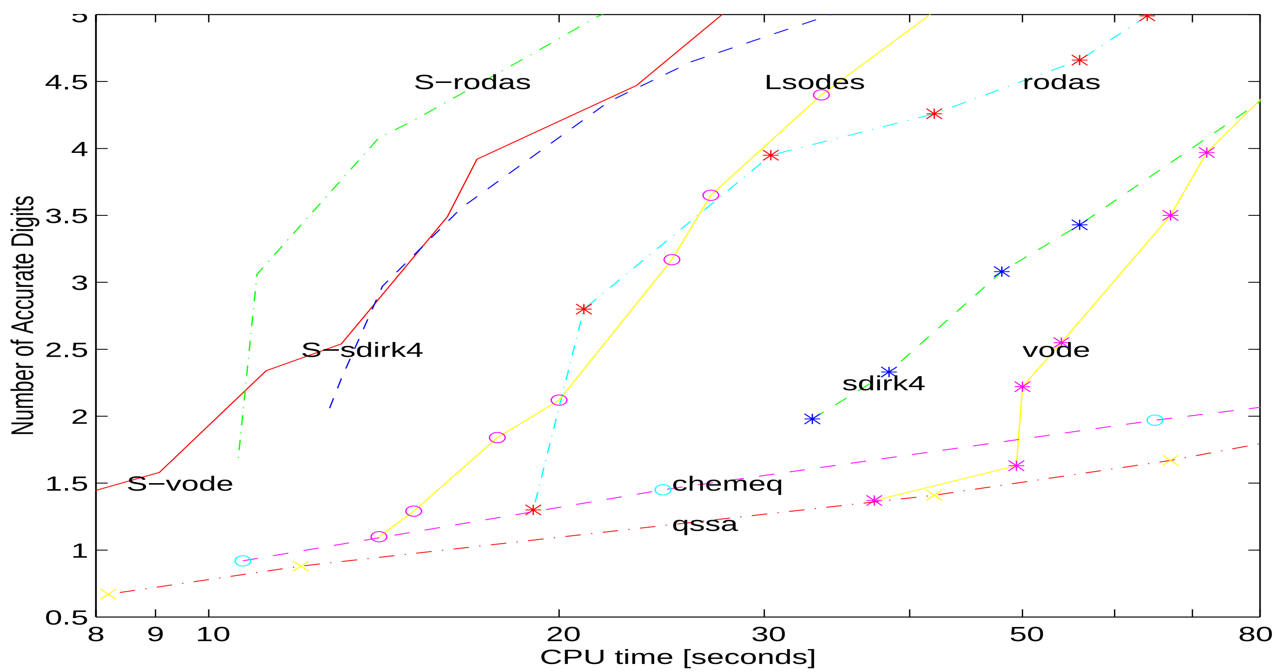
JacVar(…), JacVar_SP(…),
JacVar_SP_Vec(…), JacVarTR_SP_Vec(…)
KppDecomp(…)
KppSolve(…), KppSolveTR(…)

Computational Efficiency

Linear Algebra

(1/Lapack)	Dec	Sol	Dec+7Sol
Harwell	0.61	0.21	0.35
KPP	0.23	0.06	0.12

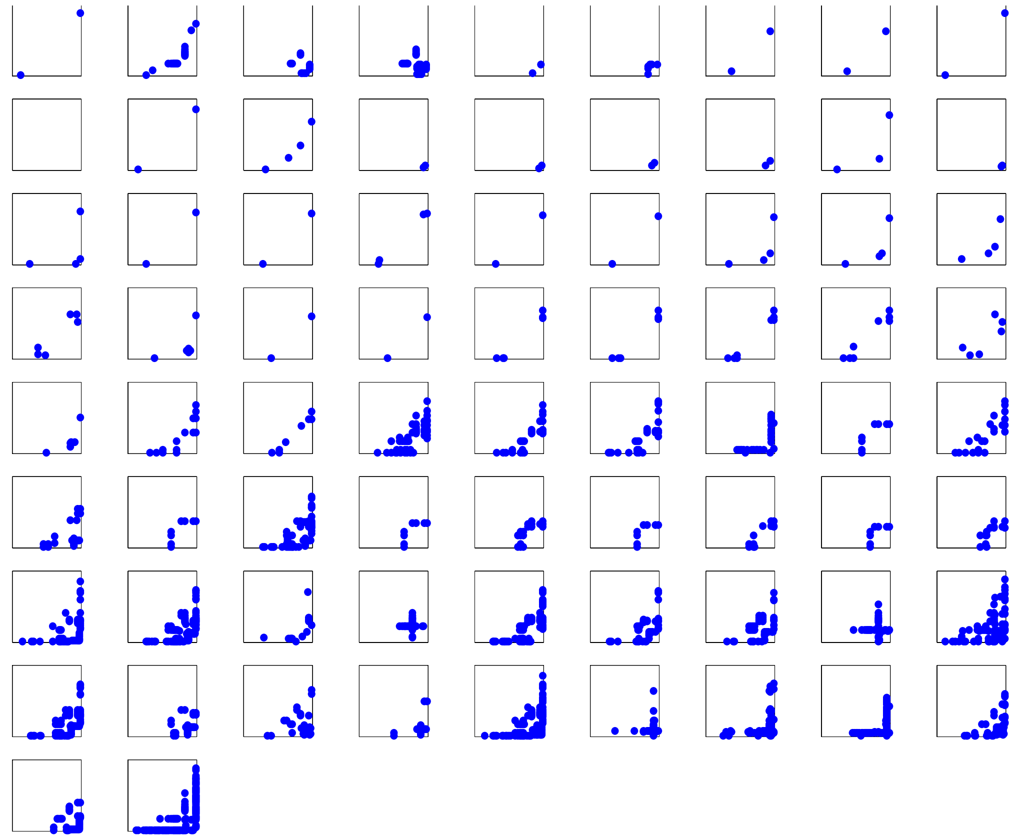
Stiff Integrators



Sparse Hessians

E.g. SAPRC99. NZ = 848x2 (0.2%)

$$H_{i,j,k} = \frac{\partial^2 f_i}{\partial y_j \partial y_k}$$



- 3-tensor
- sparse coordinate format
- account for symmetry

#HESSIAN [ON | OFF]

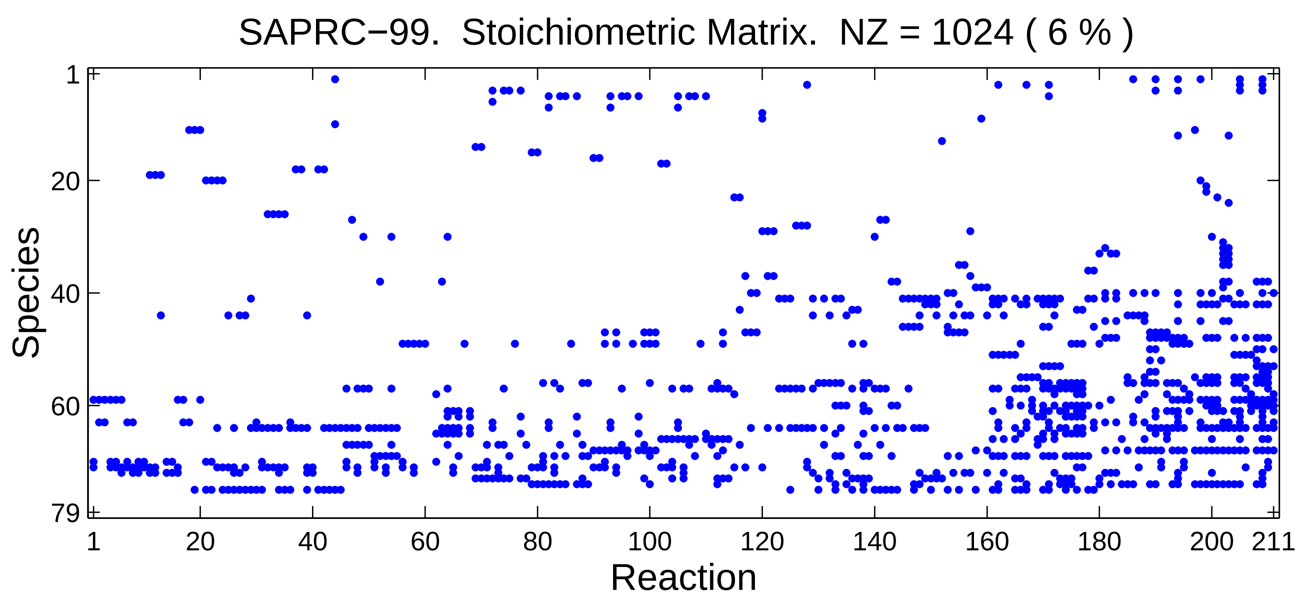
HessVar(…)
HessVar_Vec(…)
HessVarTR_Vec(…)

Stoichiometric Form

#STOICMAT [ON | OFF]

STOICM (column compressed)
ReactantProd(…)
JacVarReactantProd(…)

dFunVar_dRcoeff(…)
dJacVar_dRcoeff(…)



Requirements for Numerics

- Numerical stability (stiff chemistry)
- Accuracy: medium-low (relerr~10^{-6}-10^{-2})
- Low Computational Time
- Mass Balance
- Positivity

Stiff Integration Methods

BDF

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

Implicit Runge-Kutta

$$y^{n+1} = y^n + \sum_{j=1}^{s} b_j f\left(Y_j\right),$$

$$Y_i = y^n + \sum_{j=1}^{s} a_{i,j} f\left(Y_j\right)$$

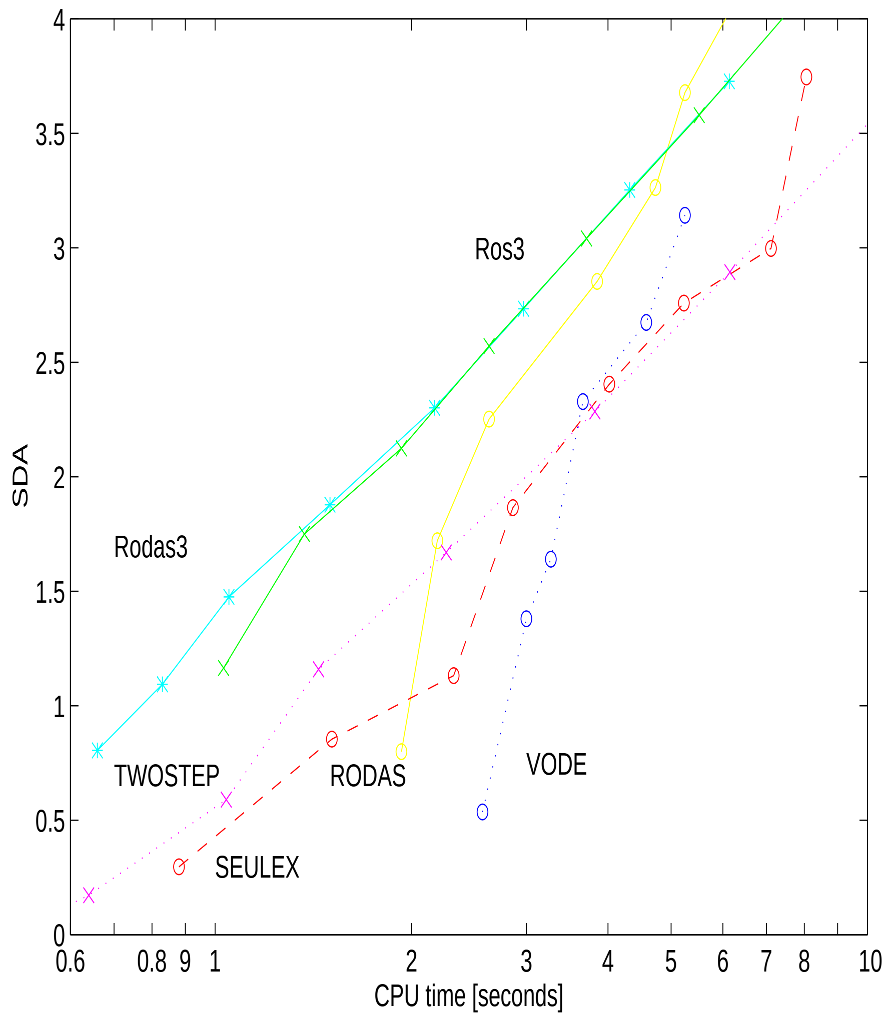
Rosenbrock Methods

$$y^{n+1} = y^n + \sum_{j=1}^{s} m_j k_j$$

$$Y_i = y^n + \sum_{j=1}^{i-1} a_{i,j} k_j$$

$$\left(\tfrac{1}{h\gamma} I - J^n\right) \cdot k_i = f\left(Y_i\right) + \sum_{j=1}^{i-1} \tfrac{1}{h} c_{i,j} k_j$$

- No Newton Iterations
- Suitable for Stiff Systems
- Mass Conservative
- Efficient: Low/Med Accuracy



Direct Decoupled Sensitivity

Problem
$$\begin{cases} y' = f(t,y,p) & S_\ell = \partial y/\partial p_\ell \\ S_\ell' = J(t,y,p)\cdot S_\ell + f_{p_\ell}(t,y,p) & 1 \le \ell \le m \end{cases}$$

$$I - h\gamma J = P^T L U \quad \Rightarrow$$

$$I - h\gamma \mathfrak{J} = \begin{bmatrix} P^T L & 0 & \cdots & 0 \\ -h\gamma\left[(JS_1)_y + J_{p_1}\right]U^{-1} & P^T L & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ -h\gamma\left[(JS_m)_y + J_{p_m}\right]U^{-1} & 0 & \cdots & P^T L \end{bmatrix} \cdot \begin{bmatrix} U & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & U \end{bmatrix}$$

DDM with KPP

BDF
(Dunker, '84)

$$y^{n+1} = \sum_{i=0}^{k-1} y^{n-i} + h\beta\, f^{\,n+1}$$

$$\left[I - h\beta\, J^{\,n+1}\right]\cdot S_\ell^{n+1} = \sum_{i=0}^{k-1} S_\ell^{n-i} + h\beta\, f_{p_\ell}^{\,n+1}, \quad 1 \le \ell \le m$$

Rosenbrock
(Sandu et. al. '02)

$$y^{n+1} = y^n + \sum_{i=1}^{s} m_i k_i^0, \quad S_\ell^{n+1} = S_\ell^n + \sum_{i=1}^{s} m_i k_i^\ell, \quad Y^i = y^n + \sum_{j=1}^{i-1} a_{ij} k_j^0$$

$$\left[\tfrac{1}{h\gamma} I - J^{\,n}\right]\cdot k_i^0 = f\left(T^i, Y^i, p\right) + \sum_{j=1}^{i-1} \tfrac{1}{h} c_{ij} k_j^0 + h\gamma_i\, f_t^{\,n}$$

$$\left[\tfrac{1}{h\gamma} I - J^{\,n}\right]\cdot k_i^\ell = J\left(T^i, Y^i, p\right)\cdot\left(S_\ell^n + \sum_{j=1}^{i-1} a_{ij} k_j^\ell\right) + f_{p_\ell}\left(T^i, Y^i, p\right)$$

$$+ \sum_{j=1}^{i-1} \tfrac{1}{h} c_{ij} k_j^\ell + J_{p_\ell}^{\,n} \cdot k_i^0 + \left(H^n \times S_\ell^n\right)\cdot k_i^0 + h\gamma_i J_t^n S_\ell^n + h\gamma_i\, f_{p_\ell,t}^{\,n}$$

Adjoint Sensitivity Analysis

Problem: Stiff ODE, scalar functional.

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

Continuous: Take adjoint of problem, then discretize.

$$\lambda' = -J^T(t,y)\cdot\lambda, \quad t^f \ge t \ge t^0, \quad \lambda(t^f) = (\nabla_y \psi)(y(t^f)) \;\Rightarrow\; \nabla_{y^0}\psi = \lambda(t^0)$$

Discrete: Discretize the problem, then take adjoint:

$$y^{k+1} = F^k(y^k), \quad k = 0,1,\cdots,N-1$$
$$\lambda^k = (\nabla_y F^k)^T(y^k), \quad \lambda^N = (\nabla_y \psi)(y^N) \;\Rightarrow\; \nabla_{y^0}\psi(y^N) = \lambda^0$$

Note: In both approaches the forward solution needs to be precomputed and stored.

Linear Multistep Methods

Method

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

Continuous Adjoint

$$\sum_{i=0}^{k} \alpha_i^{[m]} \lambda^{m+i} = h_m \sum_{i=0}^{k} \beta_i^{[m]} J^T\left(y^{m+i}\right) \cdot \lambda^{m+i}$$

Discrete Adjoint

$$\sum_{i=0}^{k} \alpha_i^{[m+i]} \lambda^{m+i} = J^T\left(y^m\right) \cdot \sum_{i=0}^{k} h_{m+i} \beta_i^{[m+i]} \lambda^{m+i}$$

Consistency: ~one-leg method, in general not consistent with continuous adj. eqn.
Implementation: with KPP

Runge Kutta Methods

Method

$$y^{n+1} = y^n + h\sum_{i=1}^{s} b_i f\left(Y^i\right), \quad Y^i = y^n + h\sum_{i=1}^{s} a_{i,j} f\left(Y^j\right)$$

Continuous Adjoint

$$\lambda^n = \lambda^{n+1} + h\sum_{i=1}^{s} b_i J^T\left(y(t^{n+1} - c_i h)\right)\cdot \Lambda^i, \quad \Lambda^i = \lambda^{n+1} + h\sum_{j=1}^{s} a_{i,j} J^T\left(y(t^{n+1} - c_i h)\right)\cdot \Lambda^j$$

Discrete Adjoint
(Hager, 2000)

$$\lambda^n = \lambda^{n+1} + \sum_{i=1}^{s} \theta^i, \quad \theta^i = hJ^T\left(Y^i\right)\cdot\left[b_i \lambda^{n+1} + \sum_{j=1}^{s} a_{j,i}\theta^j\right]$$

Consistency *(Sandu, 2003)* 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).

Implementation: with KPP

Singular Perturbation

Test problem relevant for stiff systems:

$$y' = f(y,z), \quad \varepsilon z' = g(y,z), \quad t^0 \le t \le t^f$$

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

$$\lambda(t) = \nabla_{y(t)} \psi\left(y(t^f), z(t^f)\right), \quad \mu(t) = \nabla_{z(t)} \psi\left(y(t^f), z(t^f)\right)$$

(Sandu, 2003)

If RK with invertible coefficient matrix A and R(∞) = 0;
and the cost function depends only on the non-stiff variable y

Then μ= 0 and λ are solved with the same accuracy as the original method, within O(ε).

A similar conclusion holds for continuous RK adjoints.

Rosenbrock Methods

Continuous Adjoint

$$\Lambda^i = \lambda^{n+1} + \sum_{j=1}^{i-1} a_{i,j} z_j \,, \quad Y^i = y(t^{n+1} - h\alpha_i)\,, \quad \lambda^n = \lambda^{n+1} + \sum_{j=1}^{s} m_j z_j$$

$$\left[\tfrac{1}{h\gamma} I - \left(J^{n+1}\right)^T\right] \cdot z_i = J^T\left(Y^i\right) \cdot \Lambda^i + \sum_{j=1}^{i-1} \tfrac{1}{h} c_{i,j}\, z_j \,, \quad 1 \le i \le s$$

Discrete Adjoint

$$\left[\tfrac{1}{h\gamma} I - \left(J^{n}\right)^T\right] \cdot u_i = m_i \lambda^{n+1} + \sum_{j=i+1}^{s} \left(a_{j,i} v_j + \tfrac{1}{h} c_{j,i} u_j\right)$$

$$v_i = J^T\left(Y_i\right) \cdot u_i \qquad s \ge i \ge 1$$

$$\lambda^n = \lambda^{n+1} + \sum_{i=1}^{s} \left(H^n \times k_i\right)^T \cdot u_i + \sum_{i=1}^{s} v_i$$

Consistency *(Sandu, 2003)* Similar to Runge-Kutta

Implementation: with KPP

Computational Efficiency

- $T_{discrete\ adjoint} \leq 5\ T_{forward}$ (Griewank, 2000)
- KPP/Rodas-3 *on* SAPRC-99:

$T_{cont\text{-}adj}/T_{fwd}$	$T_{cont\text{-}grad}/T_{fwd}$	$T_{discr\text{-}adj}/T_{fwd}$	$T_{discr\text{-}grad}/T_{fwd}$
1.2	3.3	2.3	4.4

KPP Numerical Library

- Simulation

  Sparse: BDF (VODE, LSODES), Runge-Kutta (Radau5), Rosenbrock (Ros-{1,2,3,4},Rodas-{3,5}. QSSA. Drivers

- Direct Decoupled Sensitivity

  Sparse: ODESSA, Rosenbrock, I.C. and R.C.

- Adjoint Sensitivity

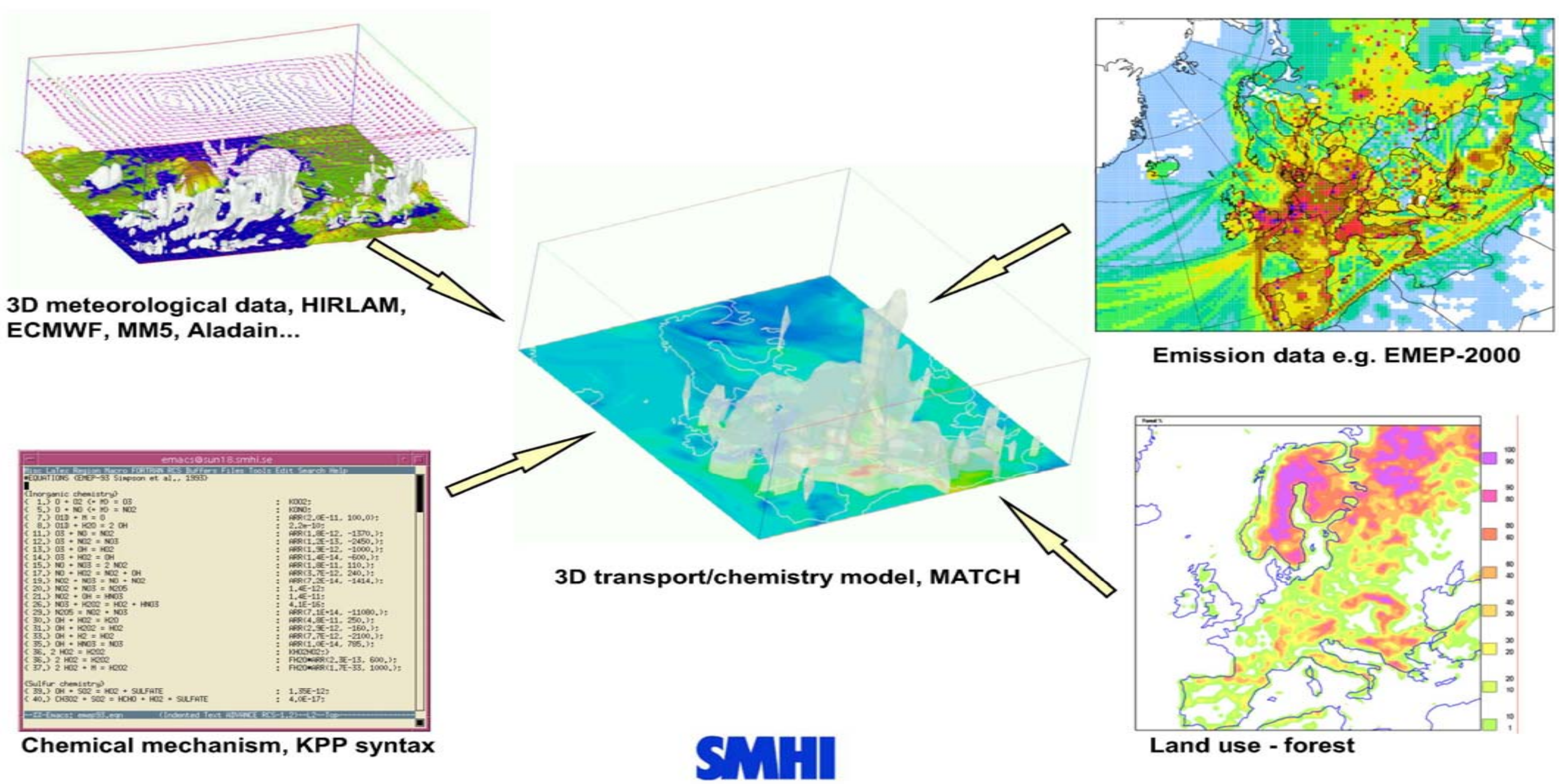
  Continuous Adj. with any simulation method

  Discrete Adj. Rosenbrock. Drivers.

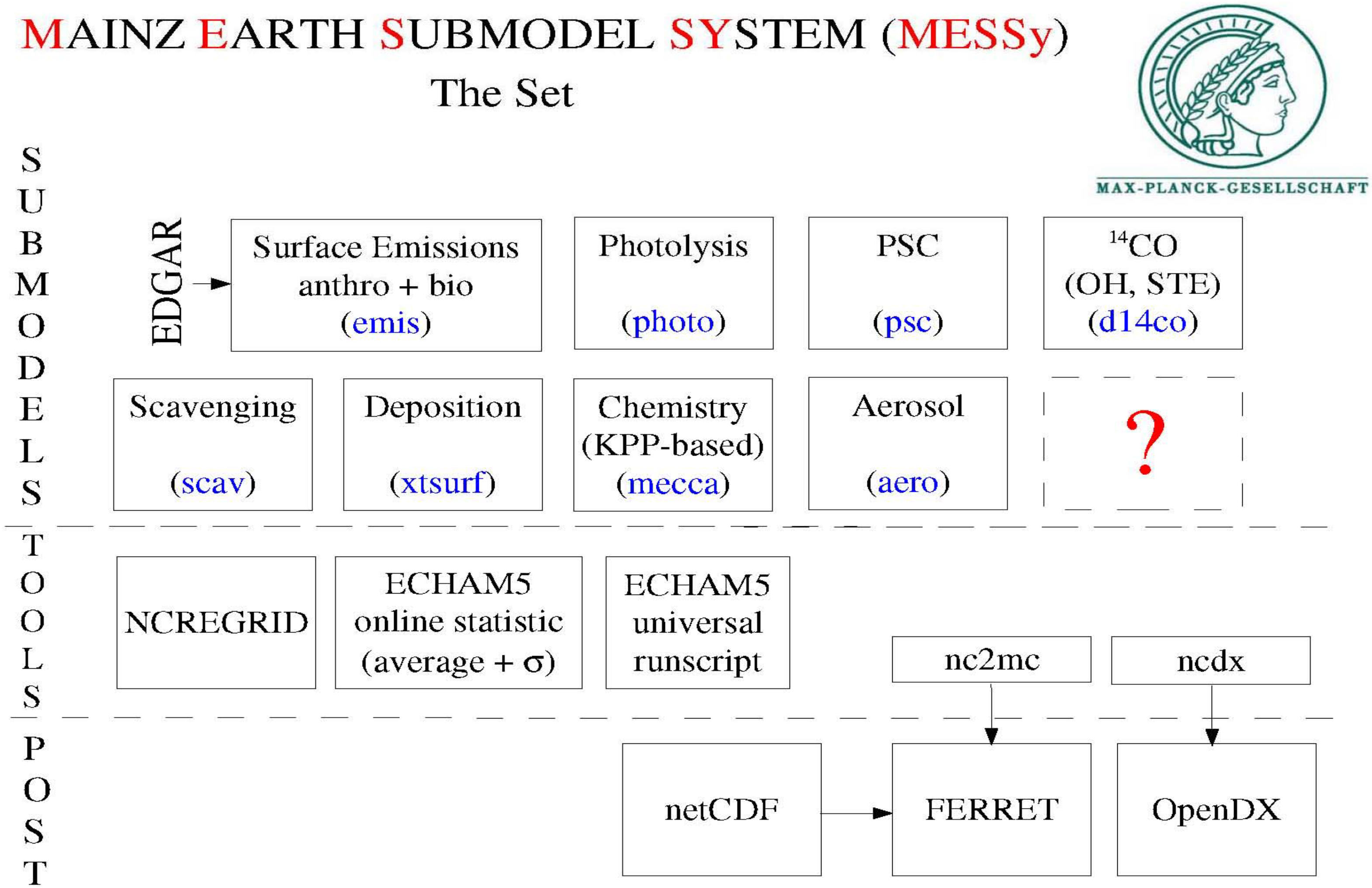
Swedish Meteo & Hydro Inst.

joakim.langner@smhi.se, **www.smhi.se**

MATCH 3D transport/chemistry/deposition model



Max Planck Institute

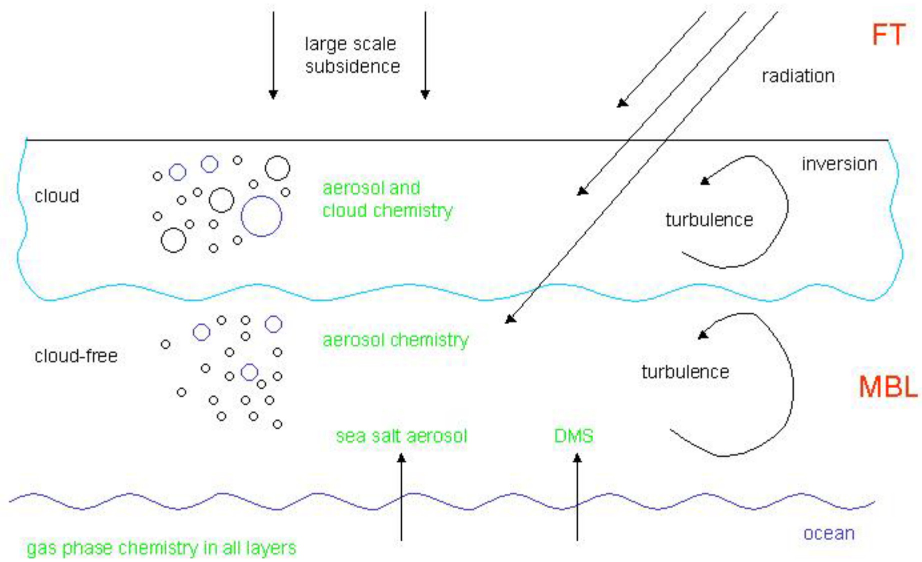


May 25, 2004

MISTRA-MPIC, U. Heidelberg

"Chemical reactions in the gas phase are considered in all model layers, aerosol chemistry only in layers where the relative humidity is greater than the crystallization humidity. […] The set of chemical reactions is solved using KPP."

(http://www.iup.uni-heidelberg.de/institut/forschung/groups/atmosphere/modell/glasow)



User Contributions to KPP

Artenum SARL
ParisCyberVillage
101-103 Bld Mac Donald
75019 Paris 19ème
France
www.artenum.com
(Consulting company)

menu... Home | The Company | Our e

Our news

Scientifc Computing News on Artenum.org

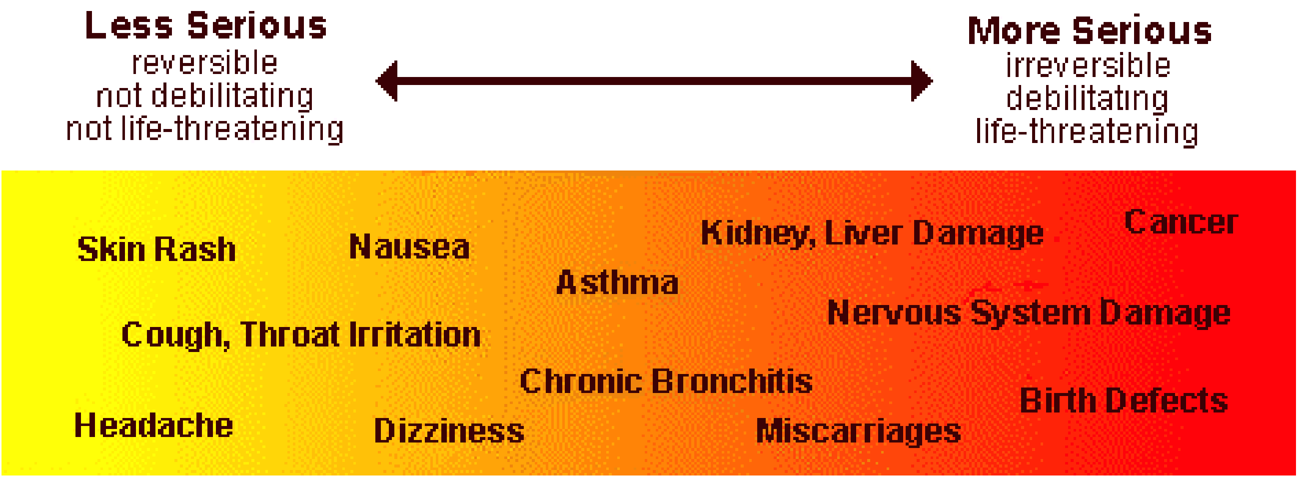
Online conference registration on Artenum.net

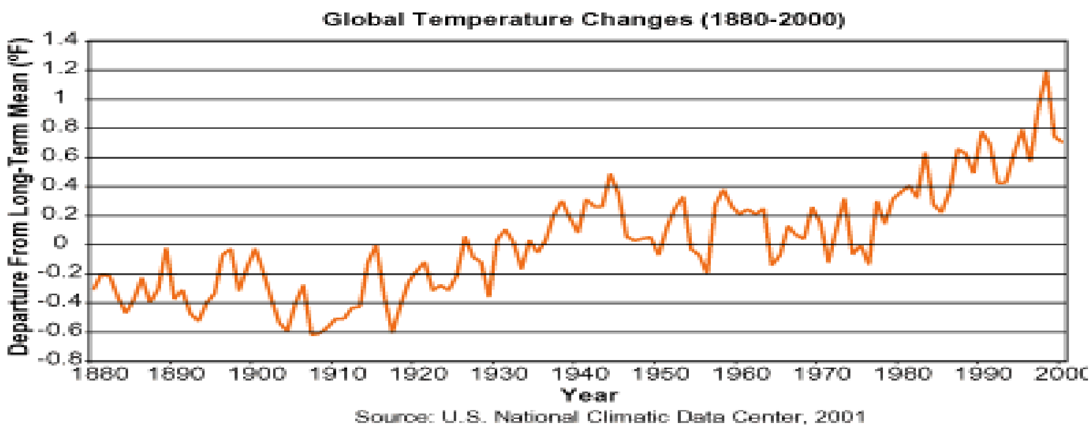
Kppcons

Download :
- the archive kppcons-1.1.3.tgz (166K)
- the patch kpp-1.1.patch.cons-1.1.3.tgz (3K)
- the technical documentation : CC_2001_06_01.pdf.gz (170K) or CC_2001_06_01.ps.gz (67K)

KPP(Kinetic PreProcessor) is a software dedicated to a quick and development of computation code for numerical simulation of chemical reaction systems. Applications fields are aerosols and modeling, combustion of hydrocarbons, atmospheric chemistry, ... It allows the definition of large set of chemical reaction systems i types of species defined by the user. But KPP doesn't take into the mass conservation law. From a mathematical point of view, it a singular jacobian matrix, restricting the use of KPP for seeking o state using Newton-Raphson algorithm.
The **Kppcons** module and the patch vanish this singularity to se fixed points by iterative Newton-Raphson method. It is used for ex interstelar chemistry.
Kppcons is written in Ansi C and is usable for all plateforms abl KPP. it is available under GPL licence.

Example: Modeling Air Pollution





The Forward Model

Mass Balance Equations. C = mole fraction. $\rho =$ air density.

$$\frac{dC_i}{dt} = -\vec{u} \cdot \nabla C_i + \frac{1}{\rho} \nabla \left(\rho K \cdot \nabla C_i \right) + \frac{1}{\rho} f_i \left(\rho C \right) + E_i$$

$$C_i \left(t^0, x \right) = C_i^0 \left(x \right), \qquad t^0 \leq t \leq t^F$$

$$C_i \left(t, x \right) = C_i^{IN} \left(t, x \right) \quad on \quad \Gamma^{IN}$$

$$K \frac{\partial C_i}{\partial n} = 0 \quad on \quad \Gamma^{OUT}$$

$$K \frac{\partial C_i}{\partial n} = V_i^{DEP} C_i - Q_i \quad on \quad \Gamma^{GROUND}$$

Discrete Forward Model

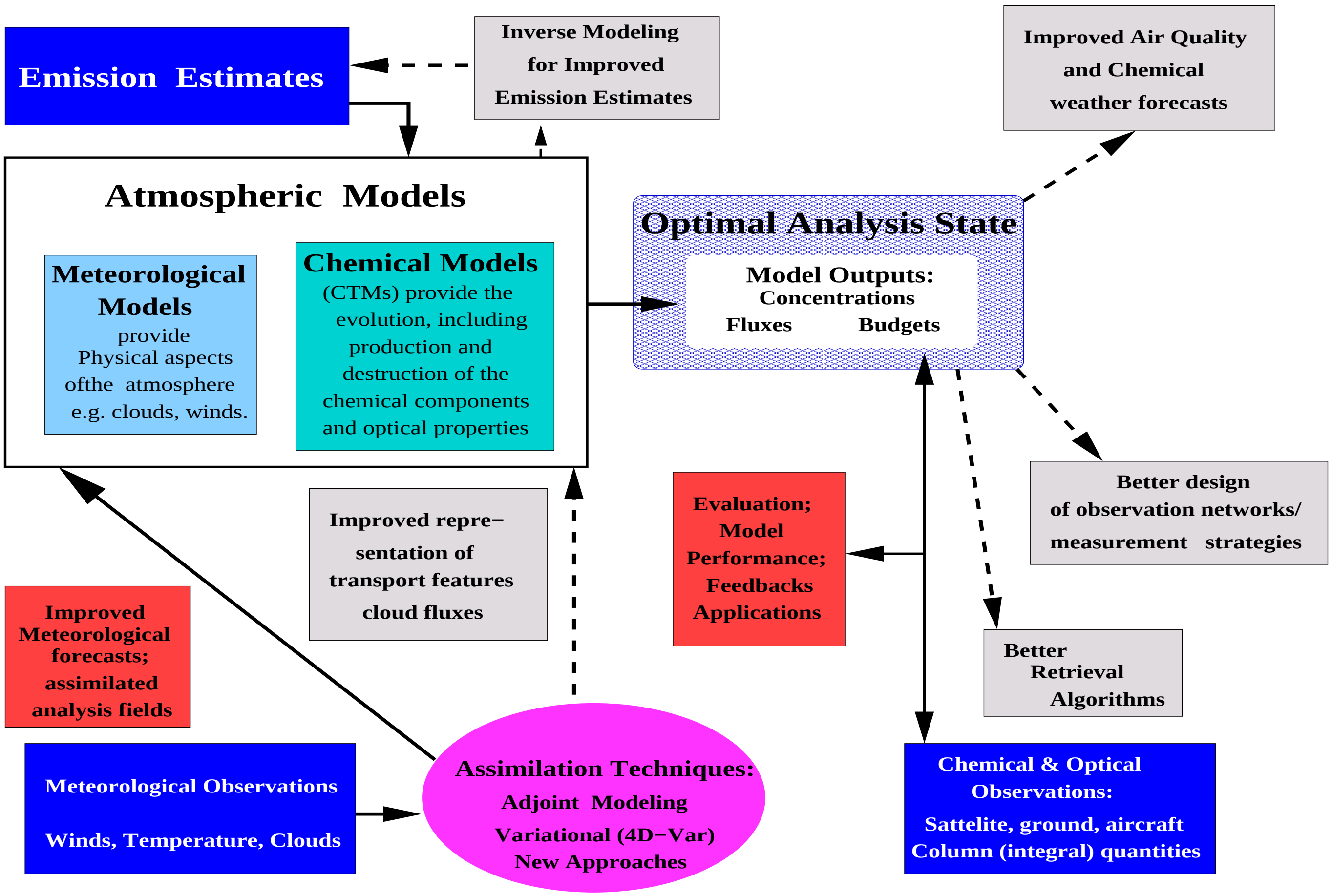
Operator Splitting:

- Conservative Methods for Transport
- Stiff Methods for Chemistry (KPP),
- Specific Methods for Aerosols,
- Different Time Steps.

$$C^{k+1} = N_{[t,t+\Delta t]} \circ C^k$$

$$N_{[t,t+\Delta t]} = T_{HOR}^{\Delta t} \circ T_{VERT}^{\Delta t} \circ R_{CHEM}^{\Delta t} \circ T_{VERT}^{\Delta t} \circ T_{HOR}^{\Delta t}$$

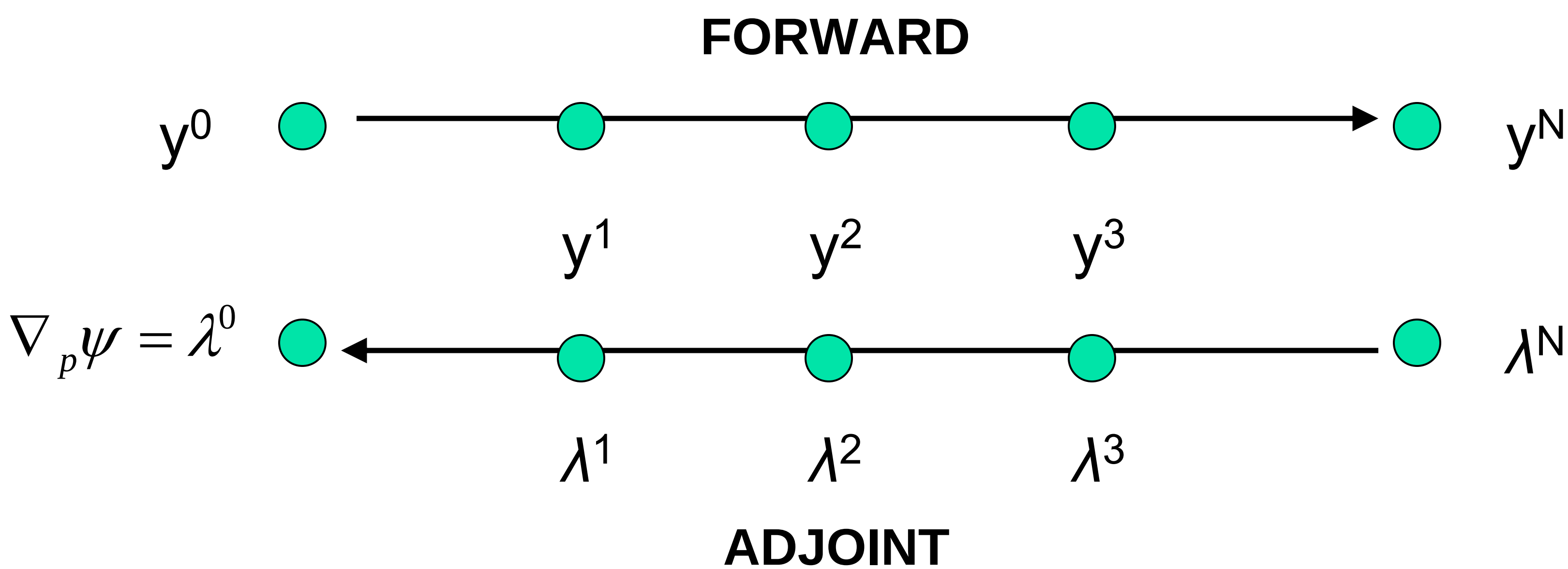
Chemical Data Assimilation



4D-Var Data Assimilation

$$\min\ \psi(p)=\tfrac{1}{2}\sum_{k=1}^{N}\left(H^k y^k - o^k\right)^T R_k^{-1}\left(H^k y^k - o^k\right)+\tfrac{1}{2}\left(p-p^b\right)^T B^{-1}\left(p-p^b\right)$$

(Note: Need the gradient of J)



Continuous Adjoint Model

$$\frac{d\lambda_i}{dt} = -\nabla\cdot(\vec{u}\lambda_i) - \nabla\cdot\left(\rho K\cdot\nabla\frac{\lambda_i}{\rho}\right) - \left(F^T(\rho C)\cdot\lambda\right)_i - \phi_i$$

$$\lambda_i\left(t^F, x\right) = \lambda_i^F(x), \qquad t^F \geq t \geq t^o$$

$$\lambda_i(t,x) = 0 \quad on \quad \Gamma^{IN}$$

$$\vec{u}\lambda_i + \rho K\frac{\partial(\lambda_i/\rho)}{\partial n} = 0 \quad on \quad \Gamma^{OUT}$$

$$\rho K\frac{\partial(\lambda_i/\rho)}{\partial n} = V_i^{DEP}\lambda_i \quad on \quad \Gamma^{GROUND}$$

Note: Linearized chemistry generated by KPP.

Discrete Adjoint Model

Operator Split Tangent Linear and Adjoint Discrete Models

$$\delta C^{k+1} = N'_{[t,t+\Delta t]} \circ \delta C^{k}$$

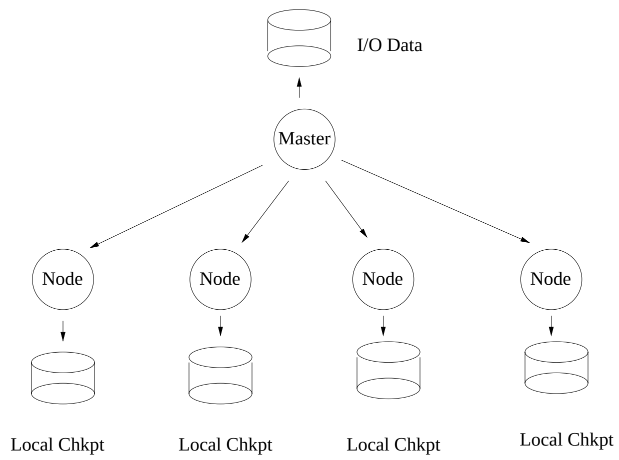
$$N'_{[t,t+\Delta t]} = \left(T_{HOR}^{\Delta t}\right)' \circ \left(T_{VERT}^{\Delta t}\right)' \circ \left(R_{CHEM}^{\Delta t}\right)' \circ \left(T_{VERT}^{\Delta t}\right)' \circ \left(T_{HOR}^{\Delta t}\right)'$$

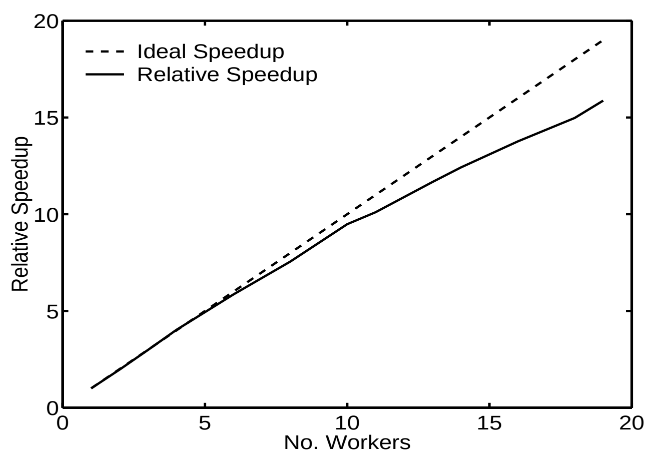
$$\lambda^{k} = N'^{*}_{[t,t+\Delta t]} \circ \lambda^{k+1}$$

$$N'^{*}_{[t,t+\Delta t]} = \left(T_{HOR}^{\Delta t}\right)'^{*} \circ \left(T_{VERT}^{\Delta t}\right)'^{*} \circ \left(R_{CHEM}^{\Delta t}\right)'^{*} \circ \left(T_{VERT}^{\Delta t}\right)'^{*} \circ \left(T_{HOR}^{\Delta t}\right)'^{*}$$

Note: Chemical Model Discrete/Continuous Adjoints automatically generated by KPP

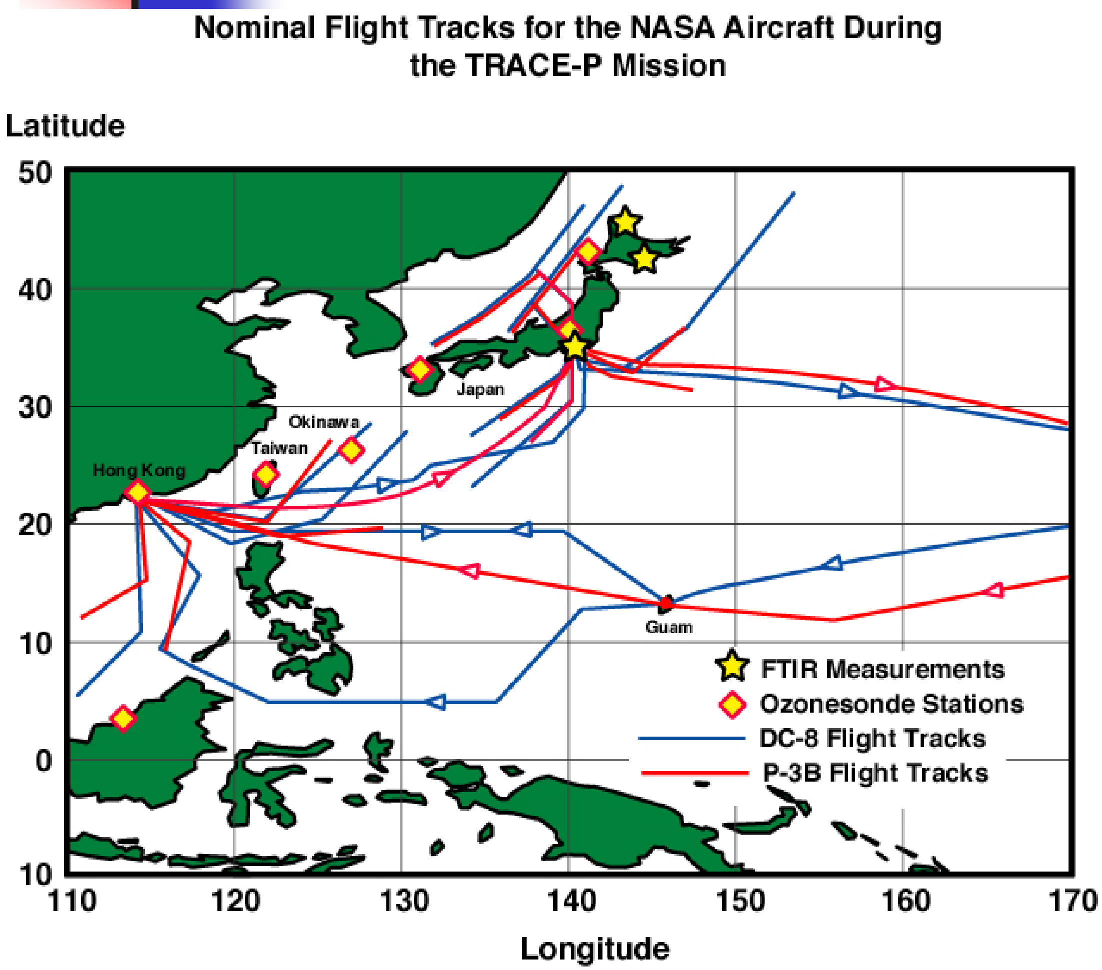
Adjoint STEM-III





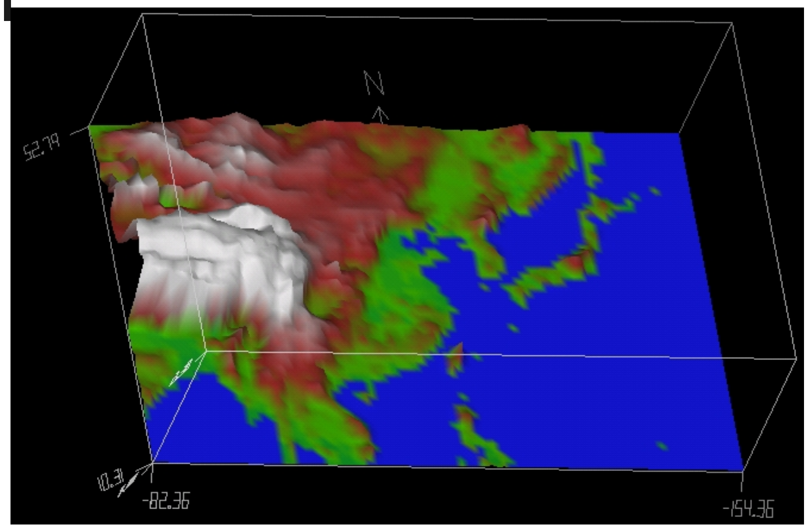
- Measurement info used to adjust initial fields and improve predictions;
- East Asia. Grid: 80 x 80Km. Time: [0,6] GMT, 03/01/01.
- SAPRC 99 (Ros-2); 3rd order upwind FD; LBFGS;
- Parallelization with PAQMSG
- Distributed Level-2 Checkpointing Scheme

NASA Trace-P Experiment

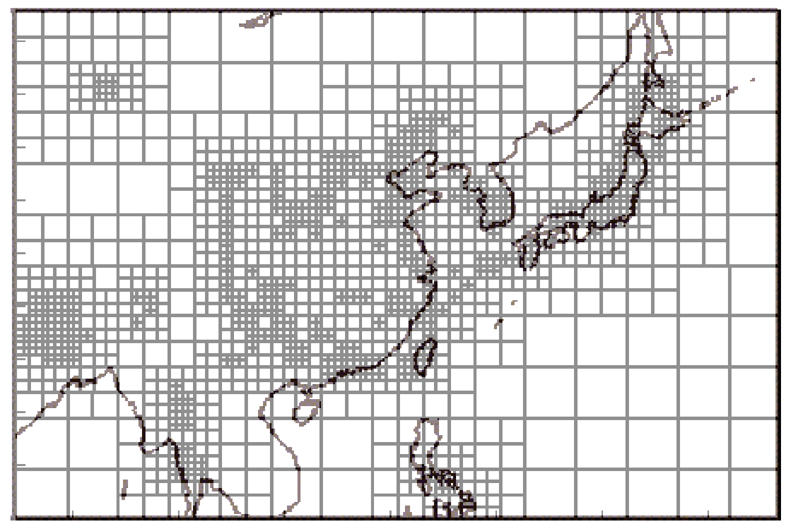


- Transport and Chemical Evolution over the Pacific
- March-April 2001
- Quantify Asian transport
- Understand chemistry over West Pacific

Computational Setting

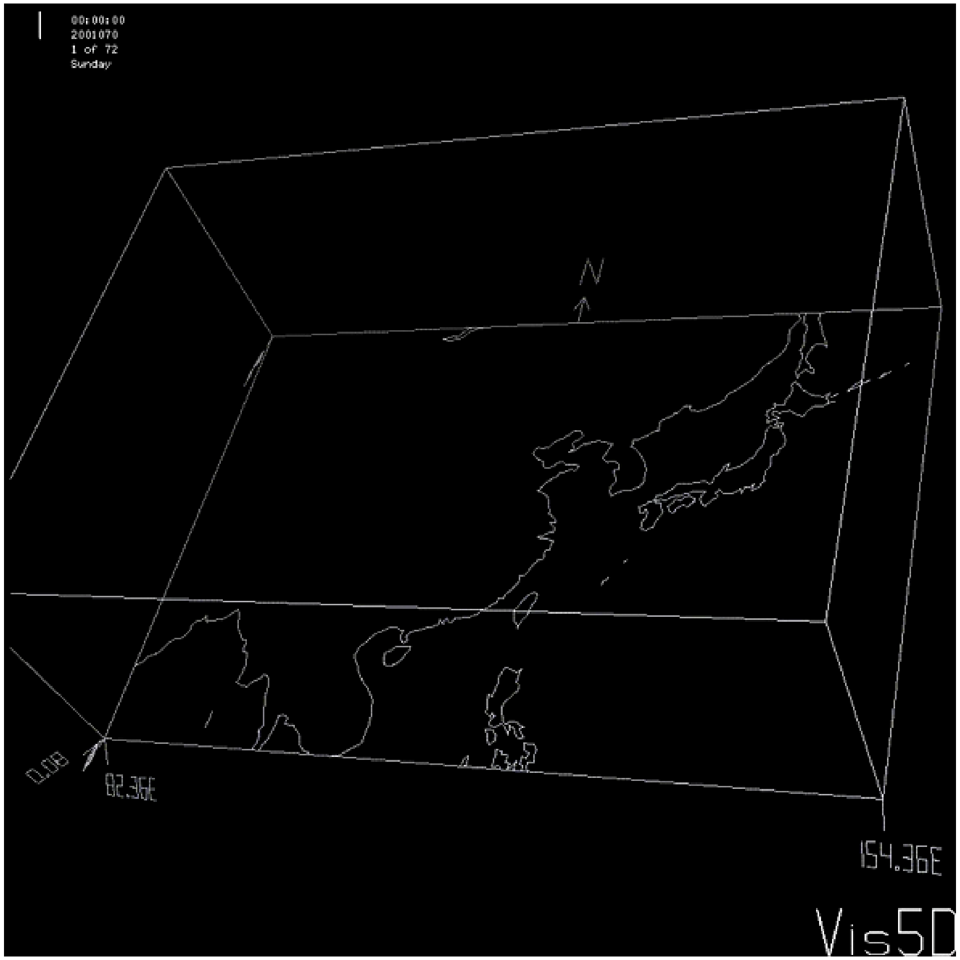


Computational domain:
- Area: 7.200 x 4.800 km
- Horizontal grid: 80 x 80 Km
- Vertical grid: 18 layers,10 km.



Model Size ~ 8,000,000 variables

Trace-P Simulation



March 01-04, 2001

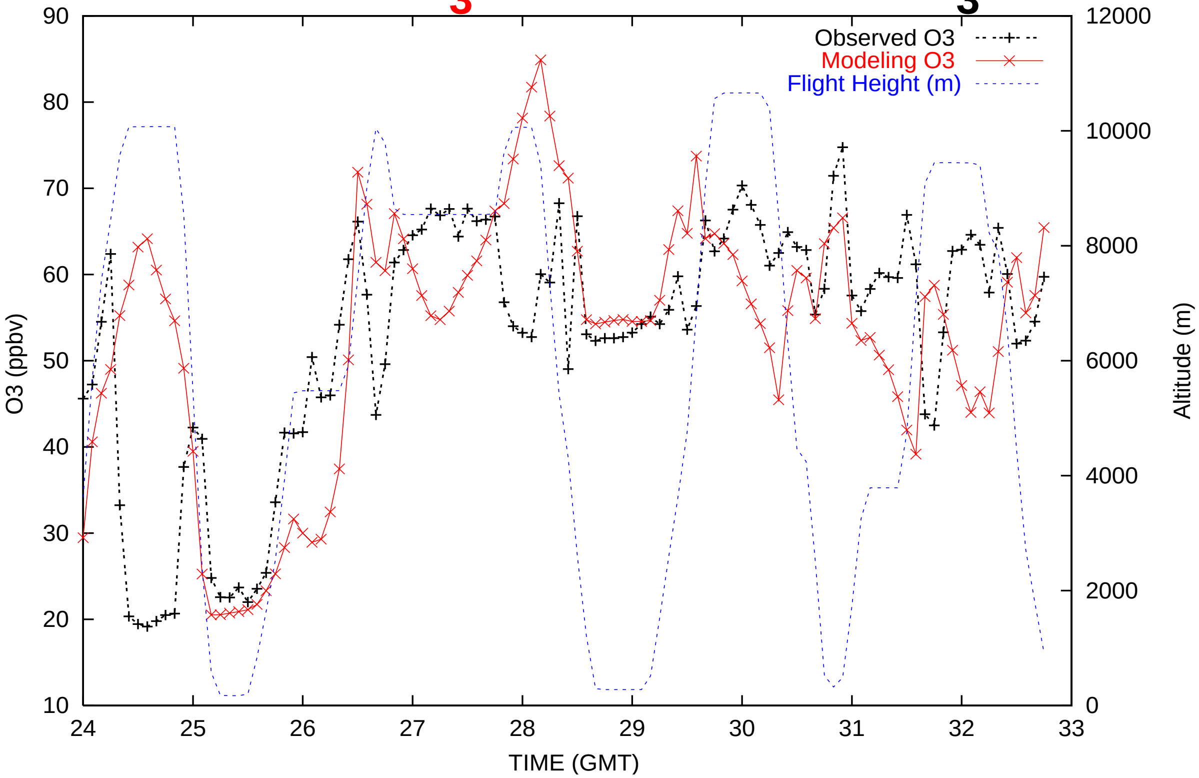
O_3

NO_2

SO_2

Model vs. Observations

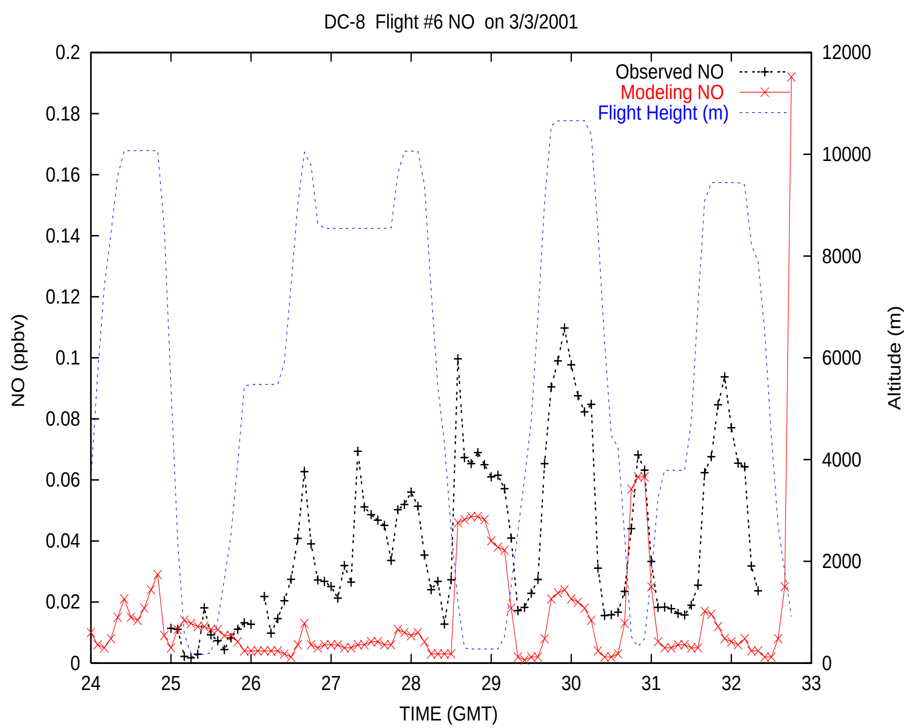
$\textbf{Modeled } O_3$ vs. $\textbf{Measured } O_3$



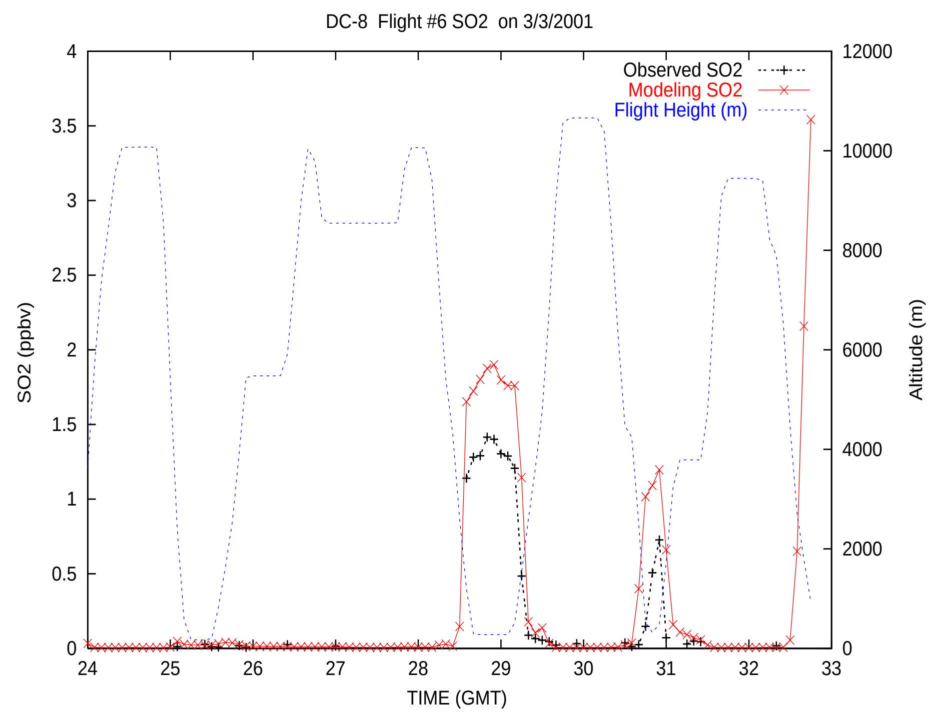
- Cost functional = model-observation gap.
- The analysis produces an optimal state of the atmosphere using:
  - ➢ Model information consistent with physics/chemistry
  - ➢ Measurement information consistent with reality

More Observations

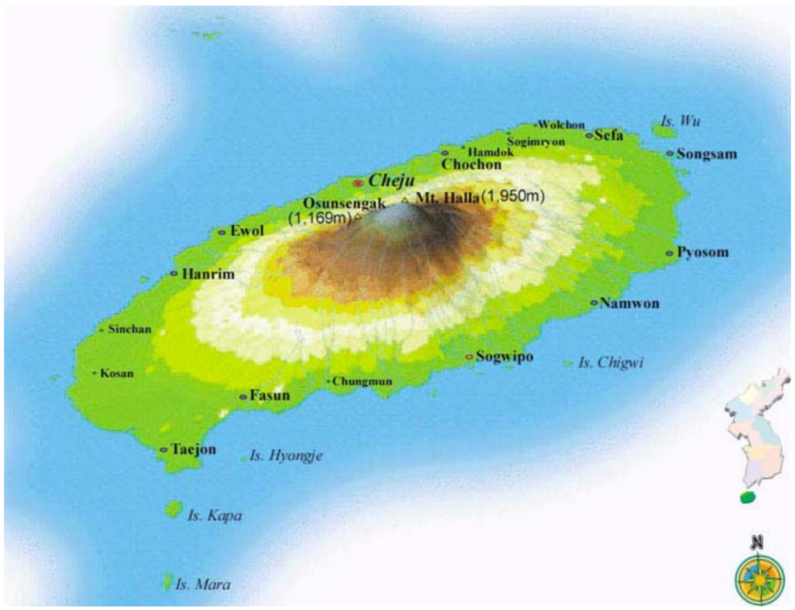
NO: Observation vs. Model



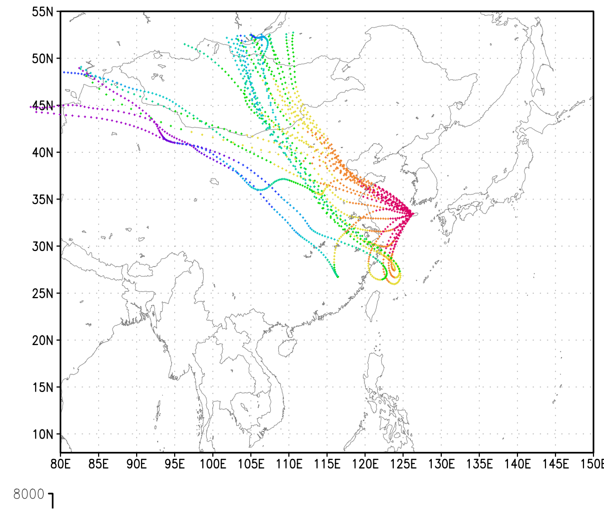
SO_2: Observation vs. Model



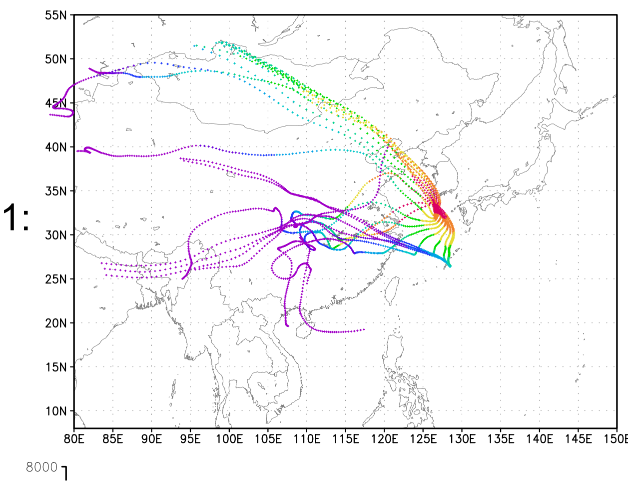
Target: O_3 at Cheju Island



March 4-6, 2001:
Strong NE flow,
Beijing-Cheju



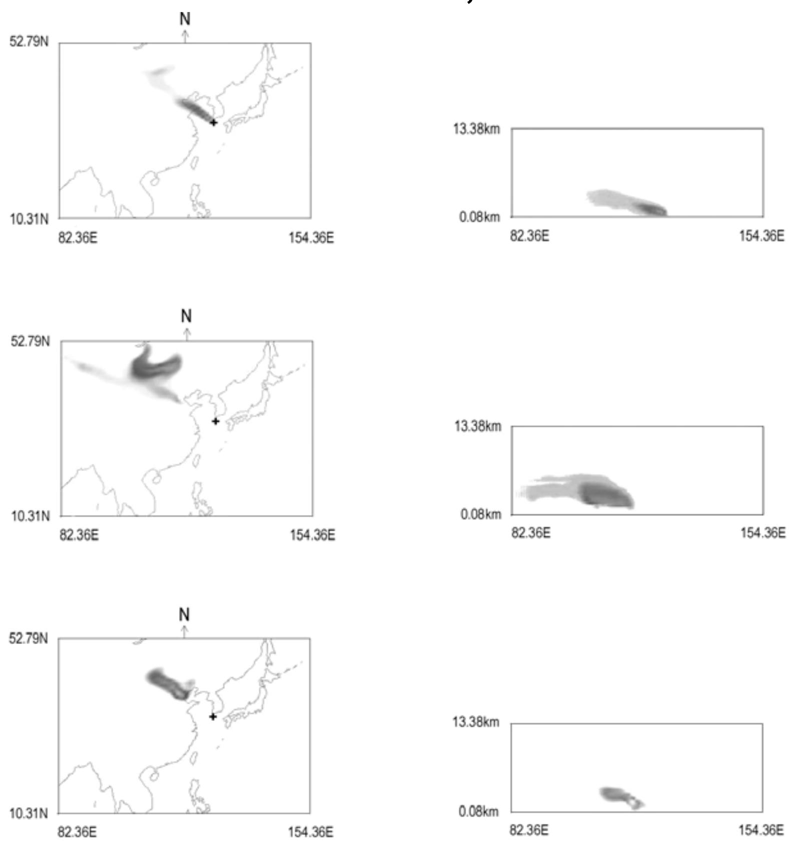
March 22-25, 2001:
Weaker Flow



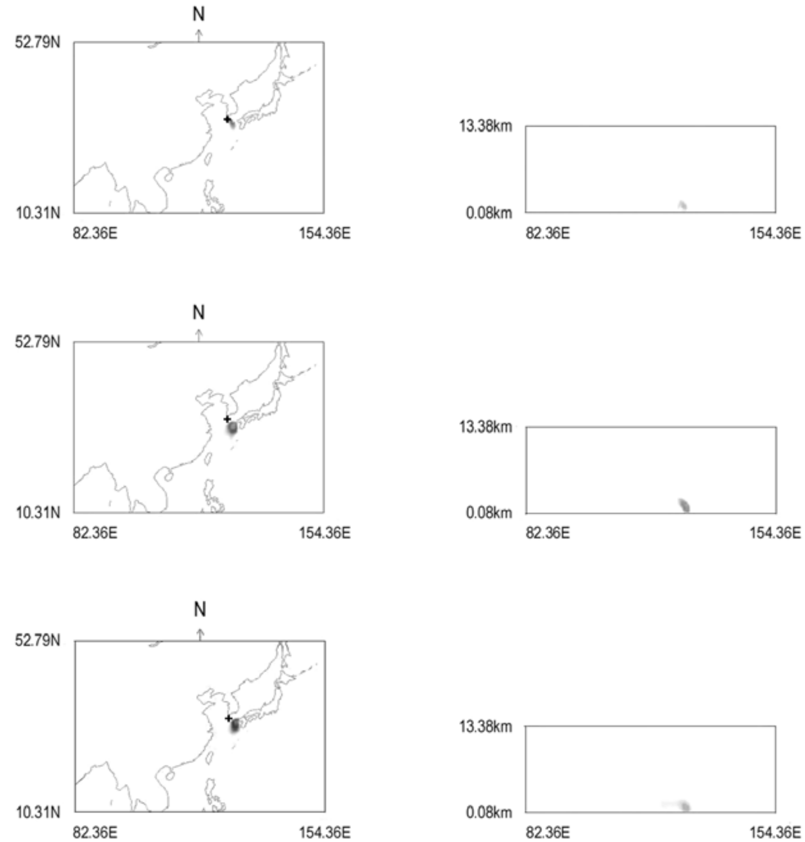
Cones of Influence

Isosurfaces of time integrals of adjoint vars. ($\psi = O_3$ at Cheju).

March 4-6, 2001



March 22-25, 2001



O_3

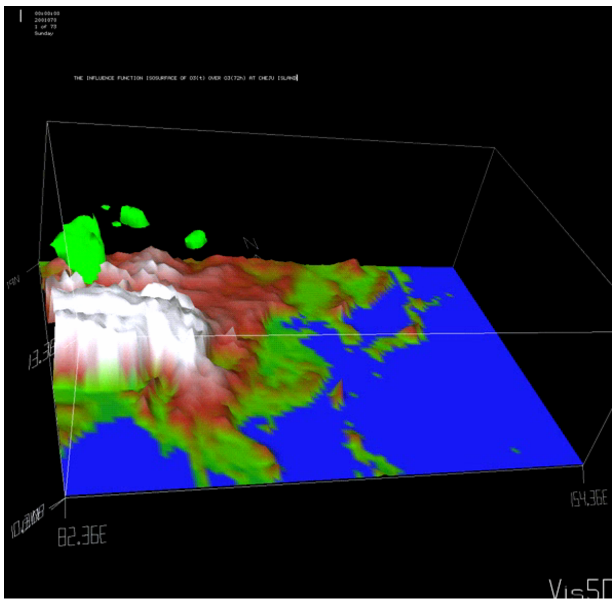
NO_2

HCHO

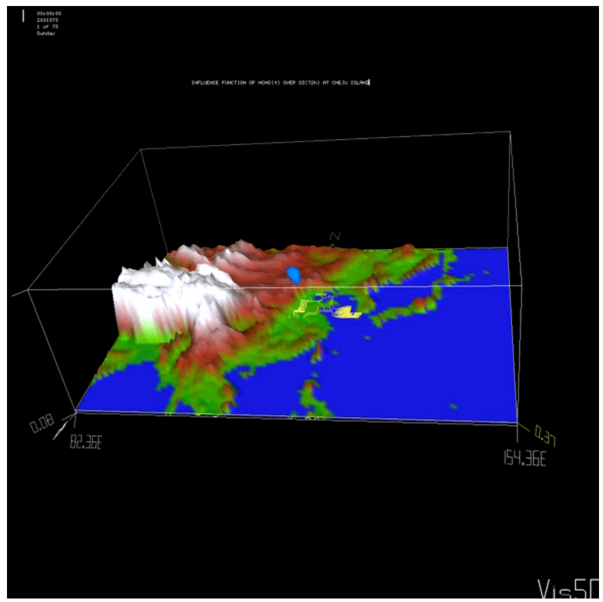
Areas of Influence

- Ψ = Ozone at Cheju, 0GMT, 03/04/2001
- Influence areas (adjoint isosurfaces) depend on meteo, but cannot be determined solely by them (nonlinear chemistry).
- Boundary condition uncertainties 3 days before.

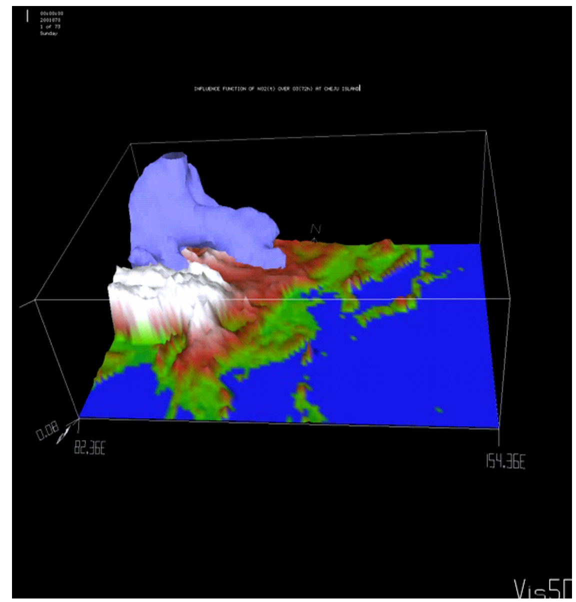
$d\,\psi / d\,O_3$



$d\,\psi / d\,HCHO$

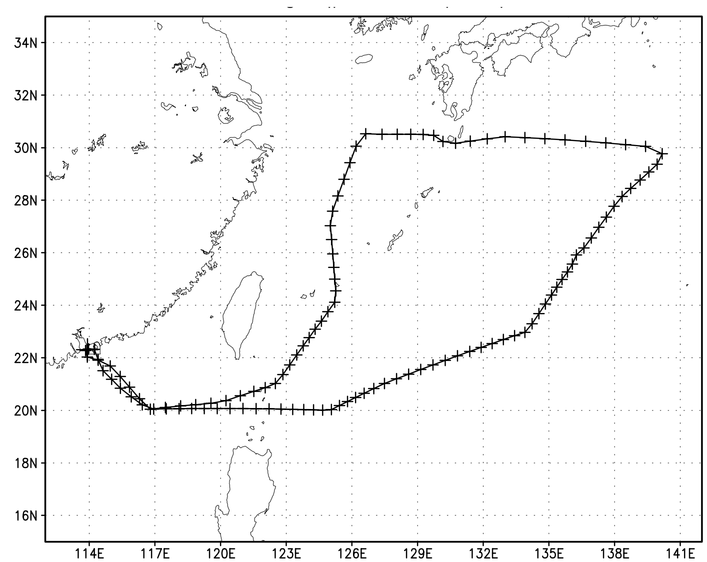


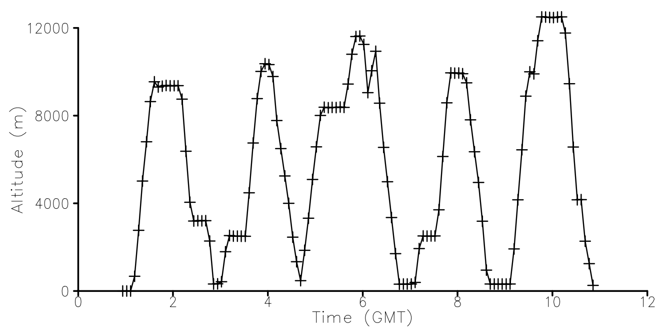
$d\,\psi / d\,NO_2$

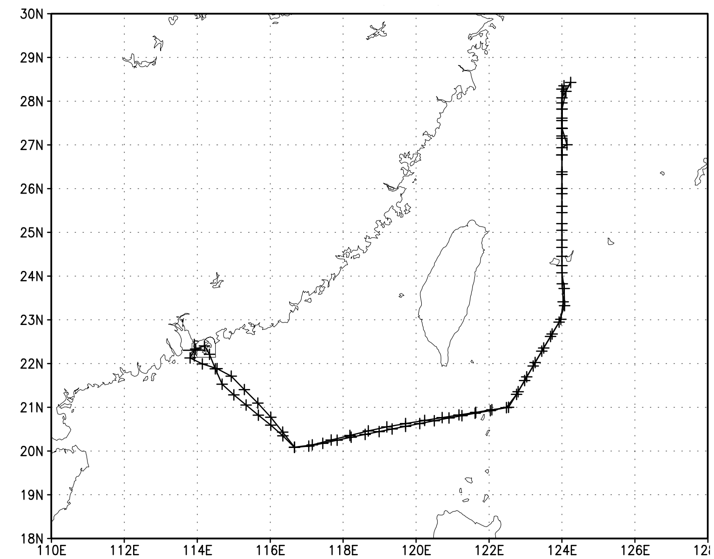


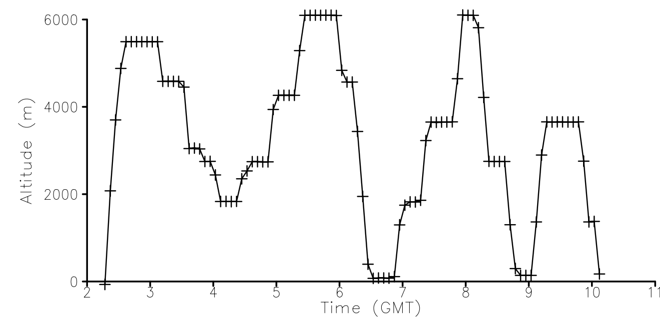
Flights on March 07, 2001

DC-8 | Taiwan (120E-122E, 22N-25N) | P3-B







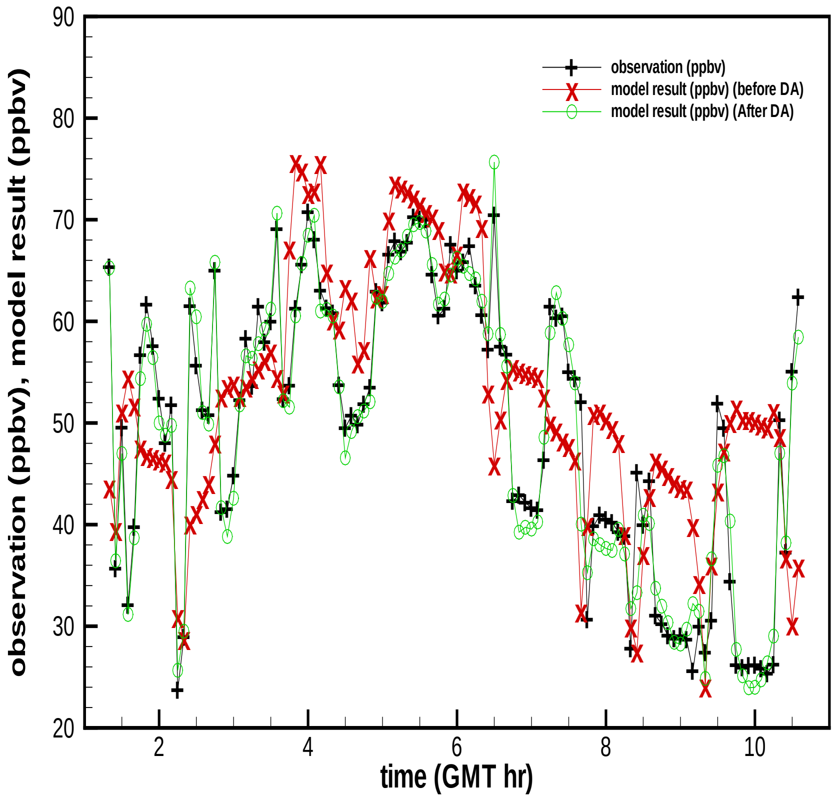


Assimilation of DC-8 O_3

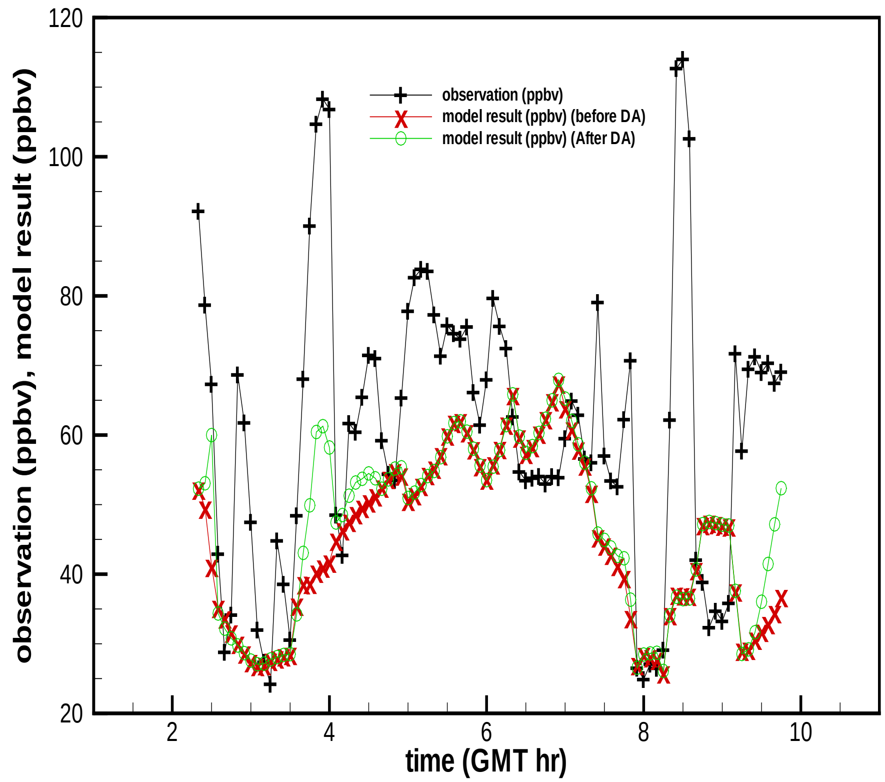
Control = Initial O_3

Assim. window [0,12] GMT

O_3 along DC-8 flight



O_3 along P3-B flight

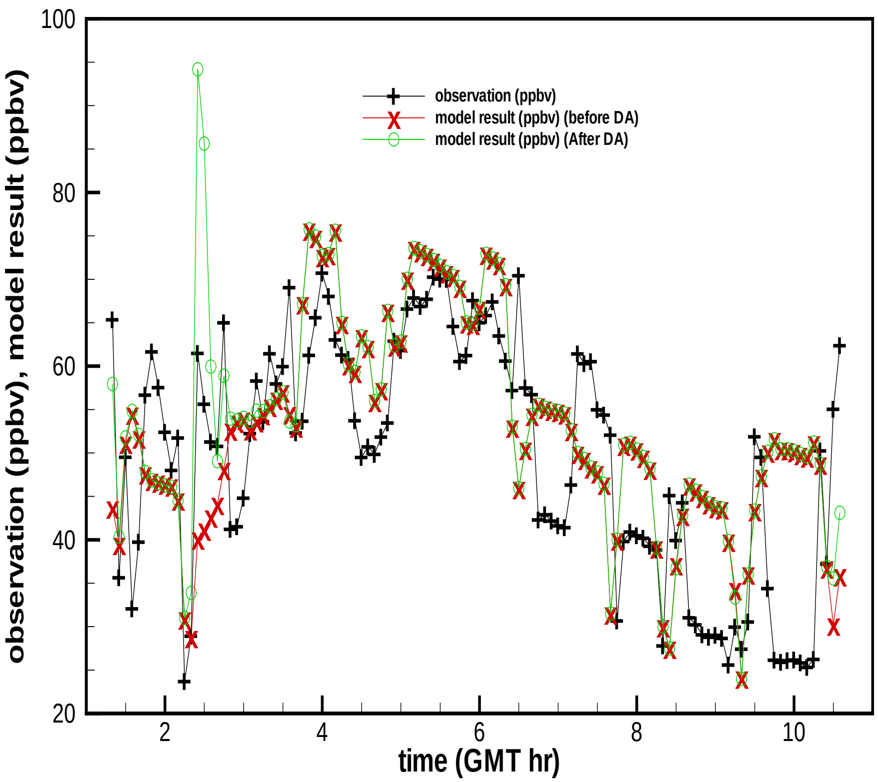


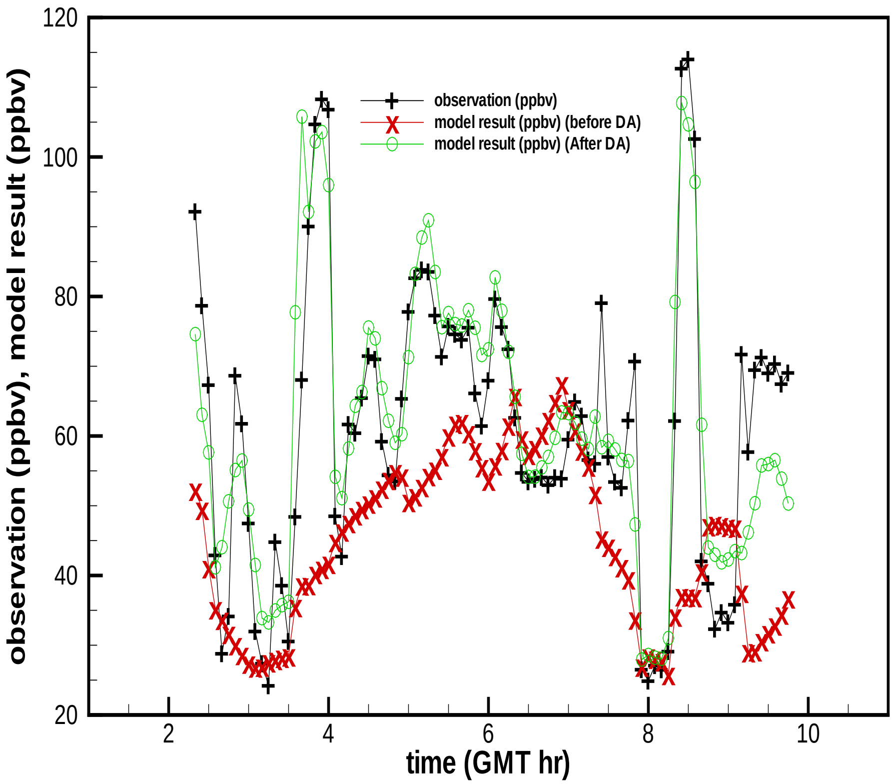
Assimilation of P3-B O_3

O_3 along DC-8 flight

Control = Initial O_3
Assim. Window [0,12] GMT

O_3 along P3-B flight

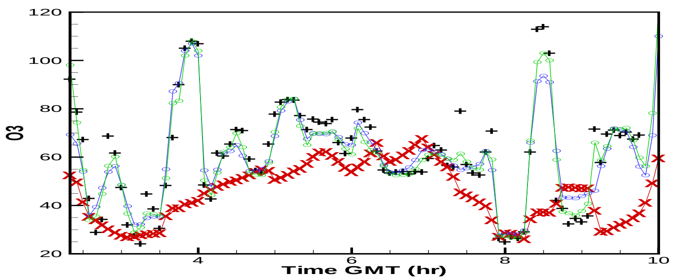




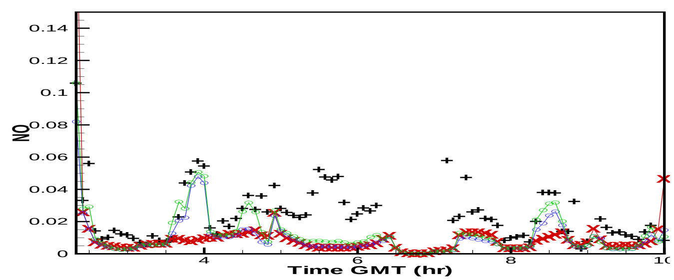
Assimilation of Multiple Species

P3-B Obs: O_3 *(8%)*; NO, NO_2 *(20%)*; HNO_3, PAN, RNO_3 *(100%)*

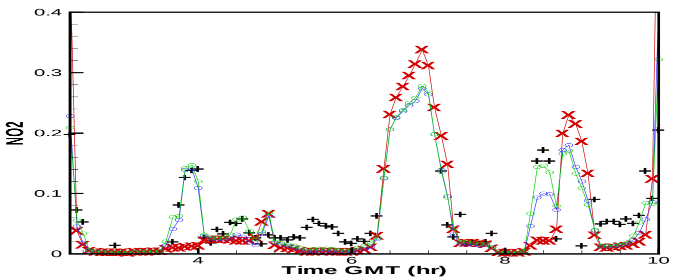
O_3



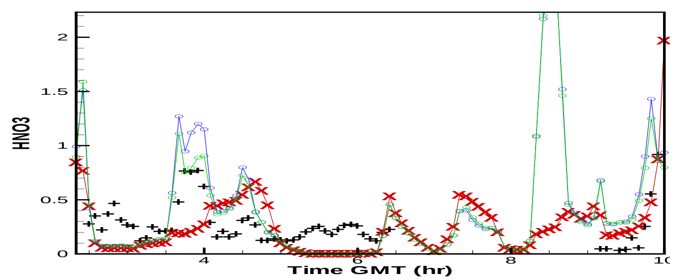
NO



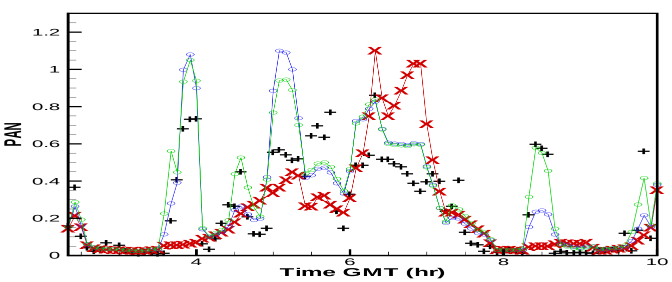
NO_2



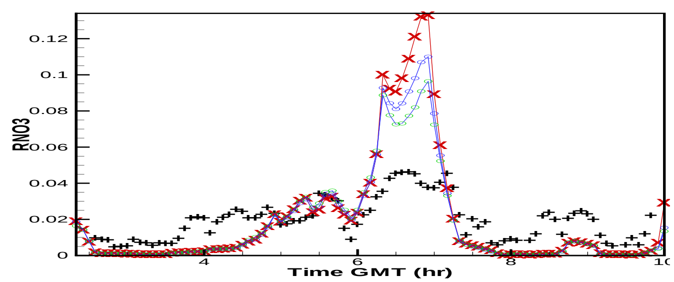
HNO_3



PAN



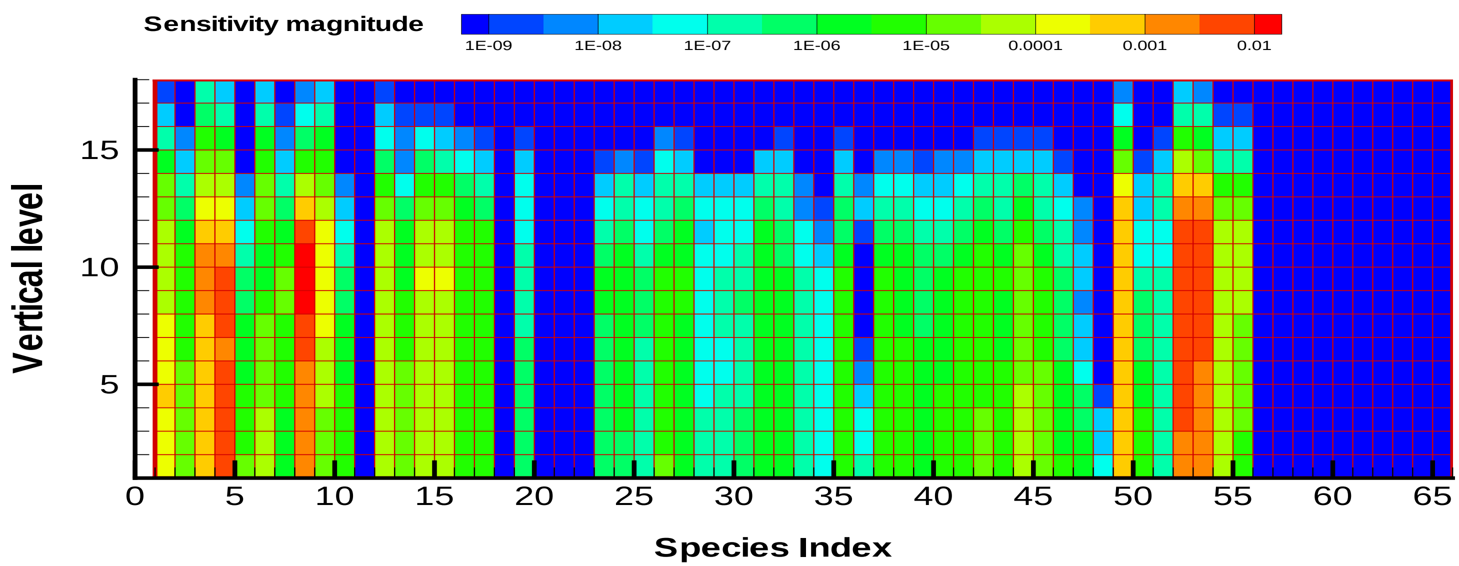
RNO_3



Sensitivity Analysis

$\psi = NO_Y$ (P3-B)

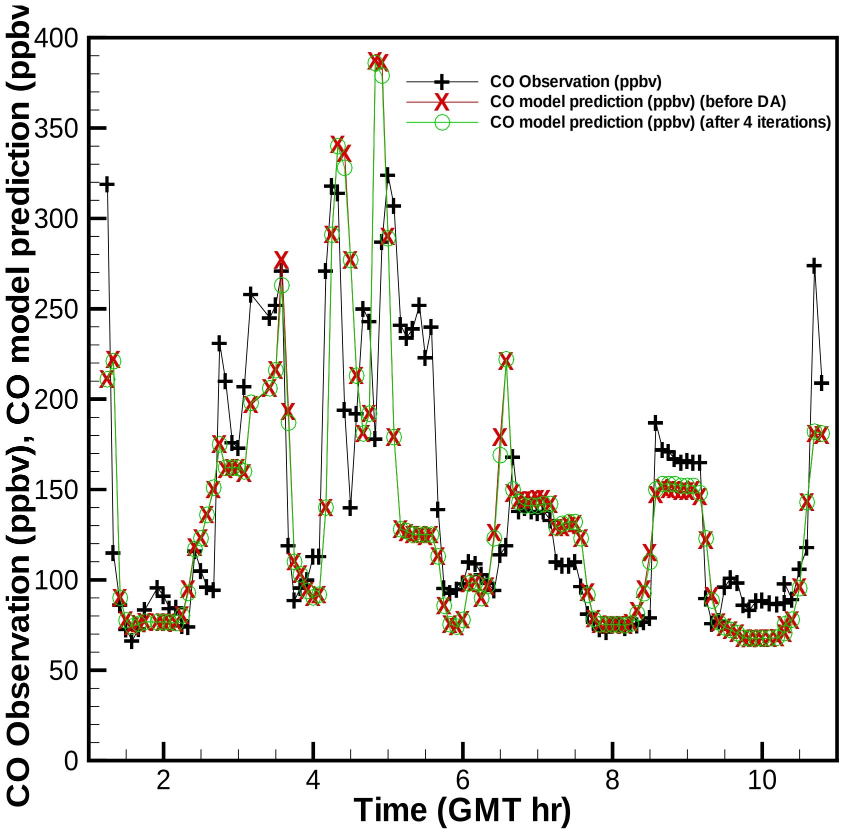
Averaged gradients help with choice of control variables



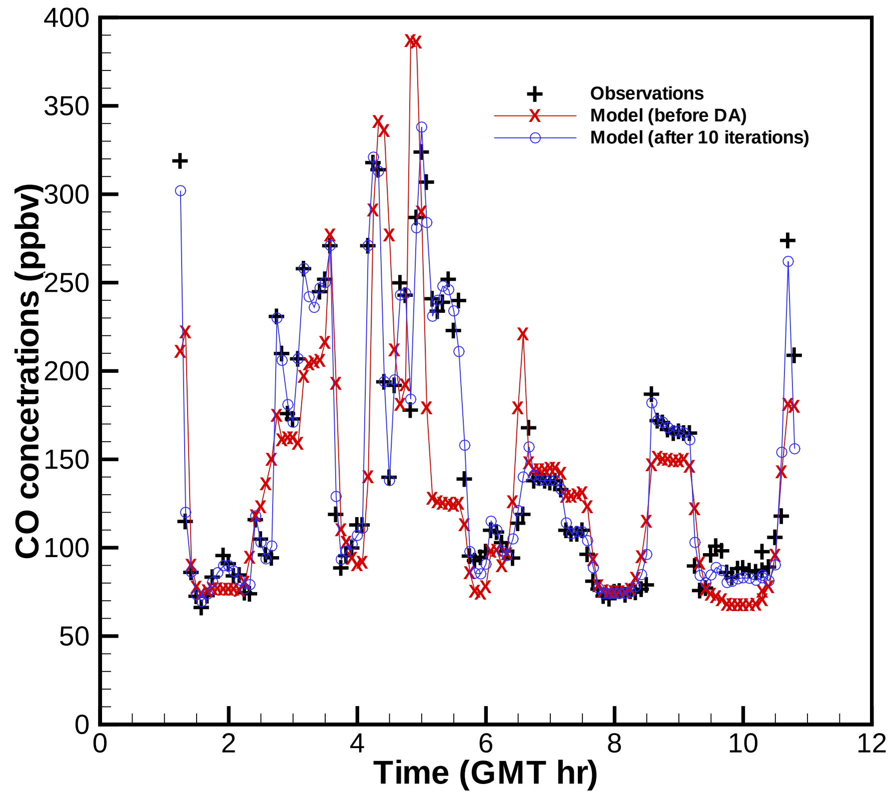
1. O3
2. H2O2
3. NO
4. NO2
5. NO3
6. N2O5
7. HONO
8. HNO3
9. HNO4
10. SO2
11. H2SO4
12. CO
13. HCHO
14. CCHO
15. RCHO
16. ACET
17. MEK
18. HCOOH
19. MEOH
20. ETOH
21. CCO_OH
22. RCO_OH
23. GLY
24. MGLY
25. BACL
26. CRES
27. BALD
28. ISOPROD
29. METHACRO
30. MVK
31. PROD2
32. DCB1
33. DCB2
34. DCB3
35. ETHENE
36. ISOPRENE
37. C2H6
38. C3H8
39. C2H2
40. C3H6
41. ALK3
42. ALK4
43. ALK5
44. ARO1
45. ARO2
46. OLE1
47. OLE2
48. TERP
49. RNO3
50. NPHE
51. PHEN
52. PAN
53. PAN2
54. PBZN
55. MA_PAN
56. BC
57. OC
58. SSF
59. SSC
60. OPM10
61. OPM25
62. DUST1
63. DUST2
64. DUST3
65. DMS
66. CO2

Assimilation of DC-8 CO

Control: Initial CO Conc.



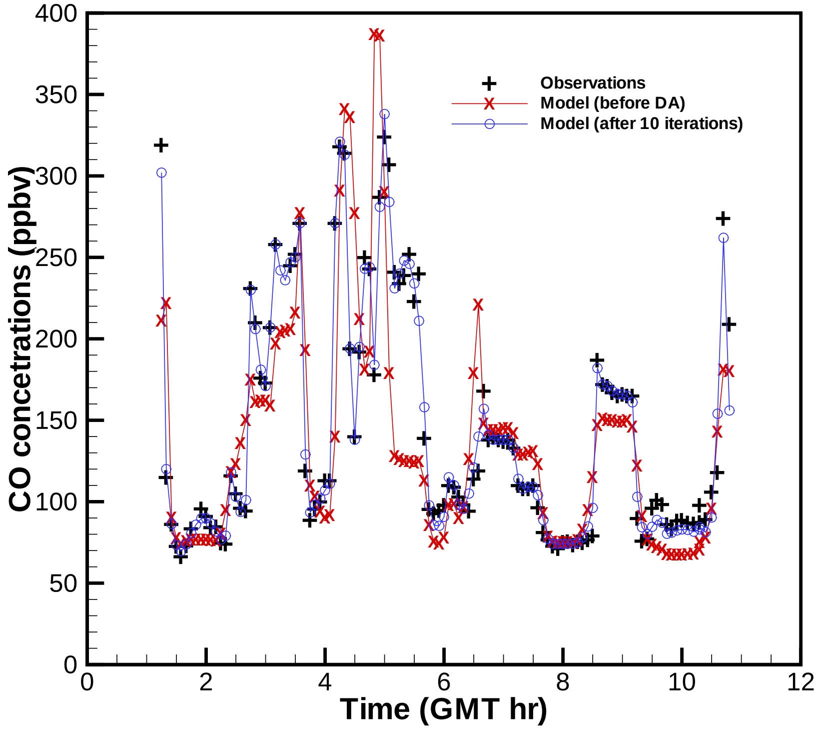
Control: Initial Conc. of 50 spc.



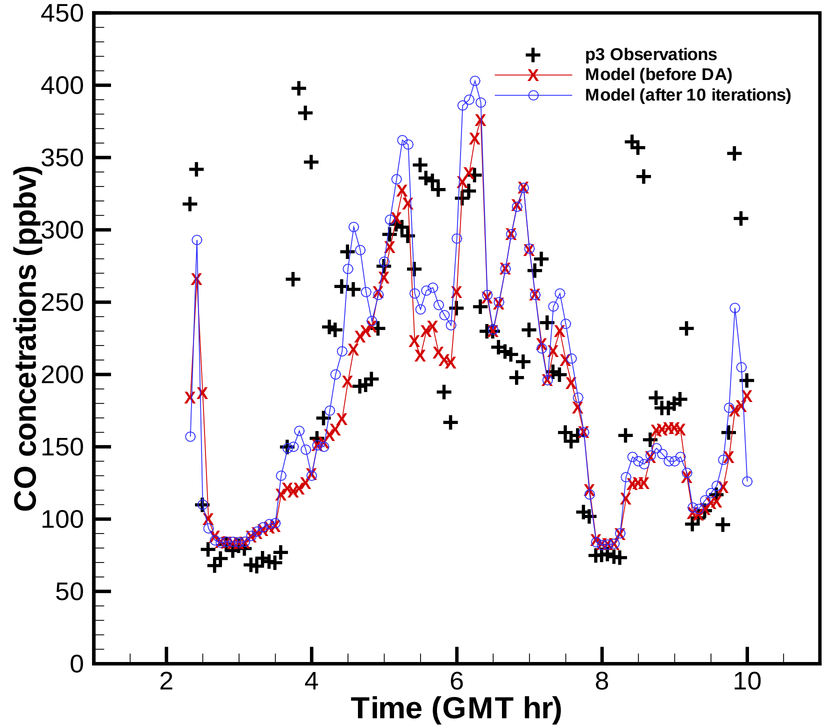
Assimilation of DC-8 CO

Control:
Initial Conc. 50 spc.

CO along DC-8 flight



CO along P3-B flight

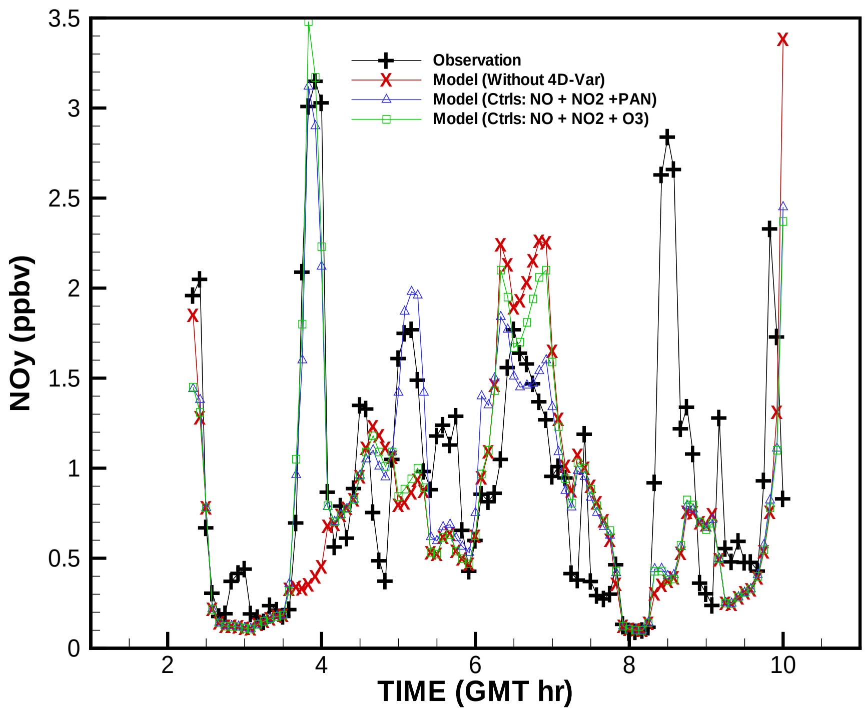


Different Control Sets

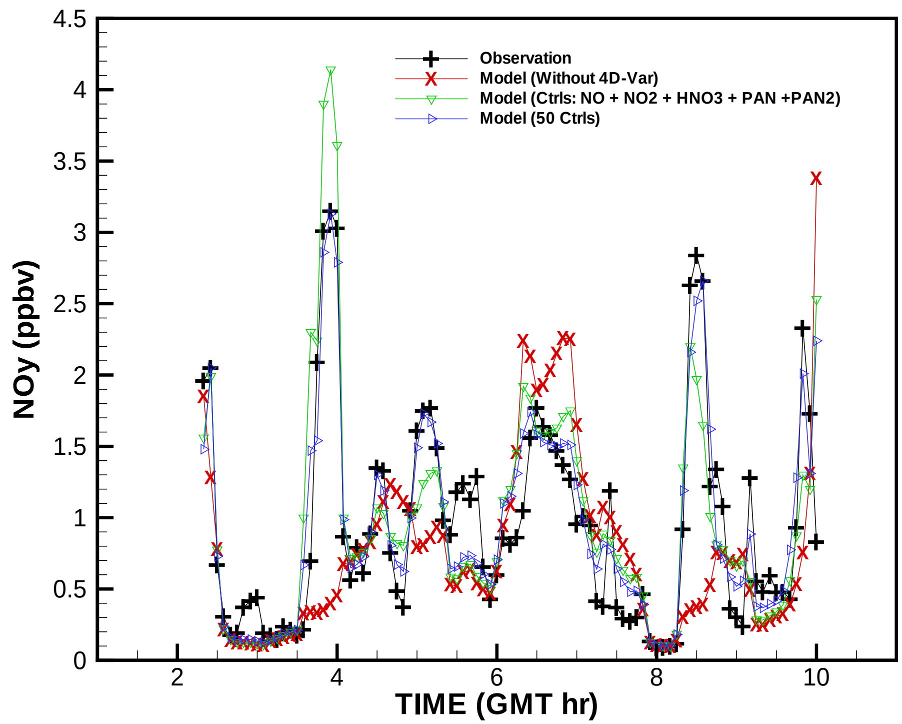
Observed: NO_Y (P3-B)

NO_Y= $NO+NO_2+NO_3+2*N_2O_5+HONO+HNO_3+HNO_4+RNO_3+PAN+PAN_2+PBZN+MA\_PAN$

(NO,NO_2,PAN) > (NO,NO_2,O_3)



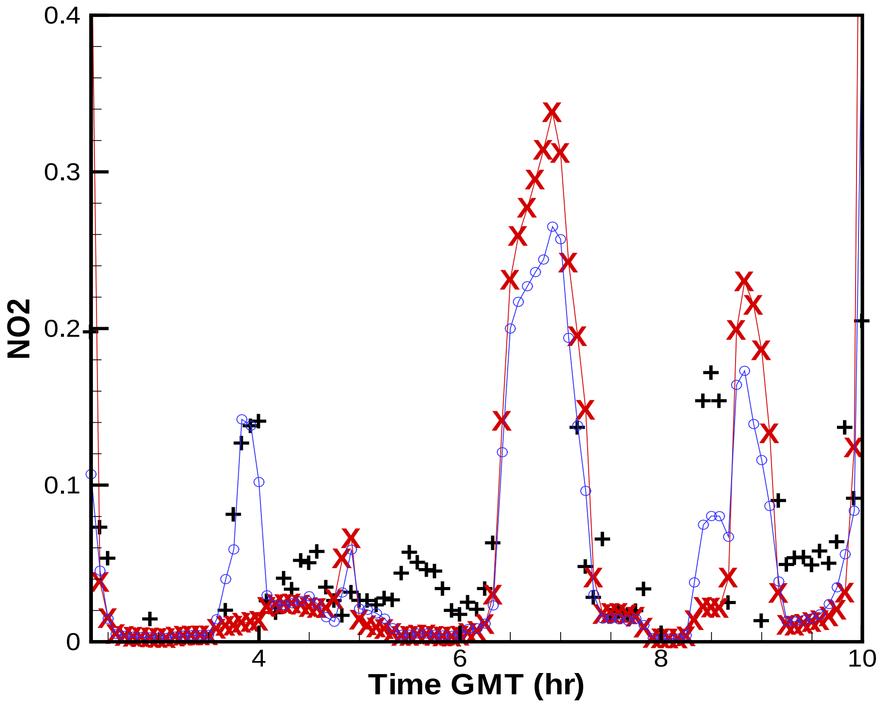
50 spc > (NO,NO_2,HNO_2,PAN,PAN_2)



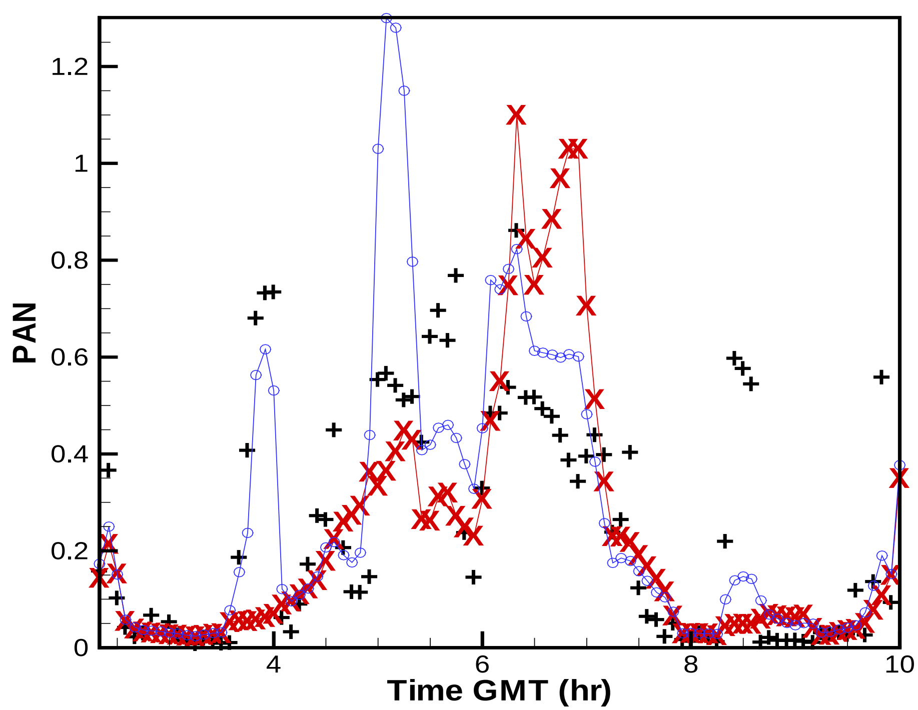
Effect on Unobserved Species

Observed: NO_Y (P3-B)
Control: Initial NO, NO_2, HNO_3, PAN, PAN_2

NO_2 (unobserved)



PAN (unobserved)



Conclusions

- **KPP** software tool for the simulation of chemical kinetics
- Code generation
- Useful (and widely used!) to build blocks for large-scale simulations
- Examples of chemical data assimilation which allows enhanced chemical weather forecasts

Quote of the Day

"Persons pretending to forecast the future shall be considered disorderly under subdivision 3, section 901 of the criminal code and liable to a fine of $250 and/or 6 months in prison."

Section 889, New York State Code of Criminal Procedure

(after M.D. Webster)

Thank you!