Ionospheric Data Assimilation: Techniques and Performance

Anthony Mannucci, JPL
Xiaoqing Pi, JPL
Attila Komjathy, JPL
Mark Butala, JPL
Chunming Wang, USC
Vardan Akopian, JPL
Brian Wilson, JPL

Based on material originally developed by Brian Wilson (JPL) for the URSI General Assembly 2008
Outline

• **Why data assimilation?**
  • To advance science, and for societal benefit

• **What is data assimilation?**
  • Formal and informal definitions

• **Kalman Filter “Alphabet Soup”**
  • Full, Reduced Rank, Limited-Correlation, Optimal Interpolation
  • Extended, Ensemble

• **JPL/USC GAIM: Validation and Applications**
  • Quantifying nowcast and forecast accuracy
  • Societal benefit: calibration, ray tracing, nav/comm effects

• **Driver Estimation**
  • 4DVAR with an adjoint
  • Ensemble Kalman filter

• **(Challenges and Future Prospects)**
  • Understanding drivers & the thermosphere
  • Toward the next solar maximum
Caveats

• Not a review of all assim. efforts
  • But will mention other efforts to the best of my ability

• Not remotely comprehensive
  • Brian had 131 slides to start

• Driven by experiences of the JPL/USC GAIM team
  • New collaboration with Utah State University team
Why data assimilation?

• **Have wealth of data**
  - Ground GPS network, GPS limb scans (COSMIC), global ionosonde network, UV limb and nadir scans (SSULI/SSUSI)
  - What if you have less data?

• **The ECMWF example** (“European Center for Medium Range Weather Forecasts”)
  - Assimilate data to improve initial state
  - Is this useful for a strongly driven system such as the ionosphere?
  - Enter: driver estimation
  - Please go to the sessions on data assimilation
  - Please Google ECMWF – wealth of technical documents
  - Also: NCEP, US version
What Is Data Assimilation?

• “Resolves” the inconsistency between data and a deterministic (time dependent) model
  • Many reasons for the inconsistency
• Nature does not do this (data assimilation is not simulating a natural process)
• Goal: as quantity of data increases, model output approaches reality
  • Not trivial. Atmospheric data assimilation models work by throwing out data. Data can help or not with forecasts.
• What about driven systems?
GPS Data as a Major Source

GPS line-of-sight TEC data collected from the IGS global GPS network are assimilated into GAIM on a daily basis and a subset of the data is assimilated in real-time.

GPS occultation and orbit data are also assimilated into GAIM as the data are available from several Earth science missions, such as CHAMP, SAC-C, IOX, COSMIC & C/NOFS!
JPL/USC GAIM: RT TEC Map & Density Slices
SPEED UP
Six-satellite COSMIC constellation
Launched March 2006

COSMIC
Ionospheric Weather Constellation

Low-Earth Orbiter

GPS

Electron Density Profile

~2500 profiles/day
Ground GPS Sparse in 3D

- Links from 100 to 1000 km
- **Ground GPS TEC network**
  - Continuous in time
  - Sparse in 3D
  - Oceans not covered
  - Profile not resolved
- **COSMIC**
  - Vertical profiles
  - Random repeat time
Ground-GPS, COSMIC and Ionosonde Coverage for Nov 21, 2008

Ground GPS (one day): dense but unevenly distributed coverage

COSMIC (one day): less dense yet evenly distributed coverage
COSMIC Coverage on Nov 21, 2008

Altitude: 780 km
Inclination: 72°
C/NOFS Coverage on Nov 21, 2008

Orbital inclination 13°, perigee of 400 km and an apogee of 850 km
STOP AT NEXT SLIDE
Day-Time (15:12) GAIM Assimilation Case on Nov 21, 2008
Comparison of HWM93 and HWM07 Using GAIM

COSMIC occultations are in the vicinity of the GR13L ionosonde

- HWM07 day-time climate HmF2/NmF2 at 15:12 UT appears to be more accurate, and this is reflected in the assimilation results (but this is not the complete story – see statistics on slide 18)
SPEED UP
Why data assimilation?
Why data assimilation?

• **Have wealth of data**
  • Ground GPS network, GPS limb scans (COSMIC),
    global ionosonde network, UV limb and nadir scans
    (SSULI/SSUSI)
Why data assimilation?

• **Have wealth of data**
  • Ground GPS network, GPS limb scans (COSMIC), global ionosonde network, UV limb and nadir scans (SSULI/SSUSI)

• **Advancing science**
  • Requires quantitatively accurate physics models
  • Challenge to the modeling community
  • Advance understanding of unmeasured drivers / couplings by “inverting” for them
  • Perform “simulations” of proposed satellites and sensor platforms (best orbit? sensor tradeoffs?)
Why data assimilation?

• Have wealth of data
  • Ground GPS network, GPS limb scans (COSMIC), global ionosonde network, UV limb and nadir scans (SSULI/SSUSI)

• Advancing science
  • Requires quantitatively accurate physics models
  • Challenge to the modeling community
  • Advance understanding of unmeasured drivers / couplings by “inverting” for them
  • Perform “simulations” of proposed satellites and sensor platforms (best orbit? sensor tradeoffs?)

• Societal Benefit
  • Space Weather monitoring: nowcast, forecast
  • Applications: calibration, ray tracing, nav/comm effects
What is data assimilation?

- **Informally:** any ionospheric “retrieval” model

- **Construct a retrieval model by selecting:**
  - An empirical or first-principles physics **model**
  - A **state representation:** grid or constrained basis set
  - An **estimation strategy:** iterative tomography, type of Kalman filter, or 3D/4DVAR
THIS STUFF IS COOL. I WISH I HAD TIME…
Recipe for an Iono. Retrieval Model: Choose one from each column

<table>
<thead>
<tr>
<th>Model</th>
<th>Representation</th>
<th>Estimation Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Time Behavior</td>
<td>2D Spherical Shell</td>
<td>Profile Adjustment</td>
</tr>
<tr>
<td>• Klobuchar</td>
<td>• Global or regional TEC mapping</td>
<td>• distance weighting</td>
</tr>
<tr>
<td>• LT Persistence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empirical Model</td>
<td>Multiple Shells / Layers</td>
<td>Iterative Tomography</td>
</tr>
<tr>
<td>• IRI, RIBG, Bent</td>
<td></td>
<td>• smoothness</td>
</tr>
<tr>
<td>Summarized Physics</td>
<td>3D Basis Functions</td>
<td>• physics constraints</td>
</tr>
<tr>
<td>• e.g. PIM</td>
<td>• Horiz: splines or spherical harmonics</td>
<td></td>
</tr>
<tr>
<td>1st-principles Iono. physics</td>
<td>• Vert: tailored EOF’s, Chapman</td>
<td></td>
</tr>
<tr>
<td>model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coupled Model</td>
<td>3D Geographic Grid</td>
<td>Full Kalman Filter</td>
</tr>
<tr>
<td>• Thermo-Iono</td>
<td>• volume elements</td>
<td>• optimal estimator</td>
</tr>
<tr>
<td>• Magneto-Iono</td>
<td>• radius, lat, Ion</td>
<td>• covariance is key</td>
</tr>
<tr>
<td></td>
<td>3D Grid in Magnetic Coordinates</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• dipole or IGRF</td>
<td>Approx. or Ensemble Kalman Filter</td>
</tr>
<tr>
<td></td>
<td>• exploit physics</td>
<td>• limit correlations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• control run time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Variational methods</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 3DVAR, 4DVAR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Driver &amp; density estimation</td>
</tr>
</tbody>
</table>
Recipe: USU Gauss-Markov Reduced Rank Kalman Filter

Model
- Local Time Behavior
  - Klobuchar
  - LT Persistence
- Empirical Model
  - IRI, RIBG, Bent
- Summarized Physics
  - e.g. PIM
- 1st-principles Iono. physics model
- Coupled Model
  - Thermo-Iono
  - Magneto-Iono

Representation
- 2D Spherical Shell
  - Global or regional
    - TEC mapping
- Multiple Shells / Layers
- 3D Basis Functions
  - Horiz: splines or spherical harmonics
  - Vert: tailored EOF's, Chapman
- 3D Geographic Grid
  - volume elements
  - radius, lat, lon
- 3D Grid in Magnetic Coordinates
  - dipole or IGRF
  - exploit physics

Estimation Strategy
- Profile Adjustment
  - distance weighting
- Iterative Tomography
  - smoothness
  - physics constraints
- Full Kalman Filter
  - optimal estimator
  - covariance is key
- Approx. or Ensemble Kalman Filter
  - limit correlations
  - control run time
- Variational methods
  - 3DVAR, 4DVAR
  - Driver & density estimation

Estimate differences!
### Recipe: USU Full Physics GAIM Ensemble Kalman Filter

#### Model
- **Local Time Behavior**
  - Klobuchar
  - LT Persistence
- **Empirical Model**
  - IRI, RIBG, Bent
- **Summarized Physics**
  - e.g. PIM
- **1st-principles Iono. physics model**
- **Coupled Model**
  - Thermo-lono
  - Magneto-lono

#### Representation
- **2D Spherical Shell**
  - Global or regional TEC mapping
- **Multiple Shells / Layers**
- **3D Basis Functions**
  - Horiz: splines or spherical harmonics
  - Vert: tailored EOF’s, Chapman
- **3D Geographic Grid**
  - volume elements
  - radius, lat, lon
- **3D Grid in Magnetic Coordinates**
  - dipole or IGRF
  - exploit physics

#### Estimation Strategy
- **Profile Adjustment**
  - distance weighting
- **Iterative Tomography**
  - smoothness
  - physics constraints
- **Full Kalman Filter**
  - optimal estimator
  - covariance is key
- **Approx. or Ensemble Kalman Filter**
  - limit correlations
  - control run time
- **Variational methods**
  - 3DVAR, 4DVAR
  - Driver & density estimation
### Recipe: JPL/USC GAIM Band-Limited Kalman Filter & 4DVAR

#### Model
- **Local Time Behavior**
  - Klobuchar
  - LT Persistence
- **Empirical Model**
  - IRI, RIBG, Bent
- **Summarized Physics**
  - e.g. PIM
- **1st-principles Iono. physics model**
- **Coupled Model**
  - Thermo-ono
  - Magneto-ono

#### Representation
- **2D Spherical Shell**
  - Global or regional TEC mapping
- **Multiple Shells / Layers**
- **3D Basis Functions**
  - Horiz: splines or spherical harmonics
  - Vert: tailored EOF’s, Chapman
- **3D Geographic Grid**
  - volume elements
  - radius, lat, lon
- **3D Grid in Magnetic Coordinates**
  - dipole or IGRF
  - exploit physics

#### Estimation Strategy
- **Profile Adjustment**
  - distance weighting
- **Iterative Tomography**
  - smoothness
  - physics constraints
- **Full Kalman Filter**
  - optimal estimator
  - covariance is key
- **Approx. or Ensemble Kalman Filter**
  - limit correlations
  - control run time
- **Variational methods**
  - 3DVAR, 4DVAR
  - Driver & density estimation
What is data assimilation?
What is data assimilation?

• **A formal definition:**
  
  • Data assimilation aims at accurate re-analysis, estimation and prediction of an unknown, true state by merging observed information into a first-principles physics model.
What is data assimilation?

• **A formal definition:**
  • Data assimilation aims at accurate re-analysis, estimation and prediction of an unknown, true state by merging observed information into a first-principles physics model.

• **Recursive Bayesian estimation (huh?)**
  • Recursive least squares with *a priori statistics* of the estimated parameters
  • A time update step (state and covariance propagation)
  • =Kalman filter
  • “How missiles reach their target”
Global Assimilative Ionospheric Model
Data Assimilation Process

Driving Forces

Physics Model

Mapping State To Measurements

4DVAR

Kalman Filter

Adjustment Of Parameters

State and covariance Forecast

State and covariance Analysis

Innovation Vector
Global Assimilative Ionospheric Model
Data Assimilation Process

Driving Forces

Physics Model

Mapping State To Measurements

Adjustment Of Parameters

State and covariance Forecast

State and covariance Analysis

Innovation Vector

Kalman Filter

4DVAR

Kalman Filter

- Recursive Filtering
- Covariance estimation and state correction
  - Optimal interpolation
  - Band-Limited Kalman filter
Global Assimilative Ionospheric Model
Data Assimilation Process

- 4-Dimensional Variational Approach
  - Minimization of cost function by estimating driving parameters
  - Non-linear least-square minimization
  - Adjoint method to efficiently compute the gradient of cost function
  - Parameterization of model “drivers”

- Kalman Filter
  - Recursive Filtering
  - Covariance estimation and state correction
  - Optimal interpolation
  - Band-Limited Kalman filter
Kalman Filter Equations

State Model (propagator)
\[ x_{k+1}^f = \Psi_k x_k^a + \varepsilon_k^q \]

Measurement Model
\( (H = \text{observation matrix}) \)
\[ m_k^o = H_k x_k^f + \varepsilon_k^o \]

Noise Model
\[ \varepsilon_k^o = \varepsilon_k^m + \varepsilon_k^r \]

Measurement:
\[ E(\varepsilon_k^m, \varepsilon_k^{mT}) = M_k \]

Representation:
\[ E(\varepsilon_k^r, \varepsilon_k^{rT}) = R_k \]

Model uncertainty:
\[ E(\varepsilon_k^q, \varepsilon_k^{qT}) = Q_k \]

Forecast state using model
\[ x_{k+1}^f = \Psi_k x_k^a \]

Produce analysis using data
\[ x_k^a = x_k^f + K_k (m_k^o - H_k x_k^f) \]

\[ K_k = P_k^f H_k^T (H_k P_k^f H_k^T + R_k + Q_k)^{-1} \]

Update covariance using data
\[ P_k^a = P_k^f - K_k H_k P_k^f \]

Propagate covariance
\[ P_{k+1}^f = \Psi_k P_k^a \Psi_k^T + Q_k \]
Kalman Filter Equations

State Model (propagator)
\[ x_{k+1}^f = \Psi_k x_k^a + \varepsilon_k^q \]

Measurement Model
(\( H = \) observation matrix)
\[ m_k^o = H_k x_k^f + \varepsilon_k^o \]

Noise Model
\[ \varepsilon_k^o = \varepsilon_k^m + \varepsilon_k^r \]

Forecast state using model
\[ x_{k+1}^f = \Psi_k x_k^a \]

Produce analysis using data
\[ x_k^a = x_k^f + K_k \left( m_k^o - H_k x_k^f \right) \]
\[ K_k = P_k^f H_k^T \left( H_k P_k^f H_k^T + R_k + Q_k \right)^{-1} \]

Update covariance using data
\[ P_k^a = P_k^f - K_k H_k P_k^f \]

Propagate covariance
\[ P_{k+1}^f = \Psi_k P_k^a \Psi_k^T + Q_k \]
Kalman Filter Equations

State Model (propagator)
\[ x_{k+1}^f = \Psi_k x_k^f + \epsilon_k^q \]

Measurement Model
\( \text{H = observation matrix} \)
\[ m_k^o = H_k x_k^f + \epsilon_k^o \]

Noise Model
\[ \epsilon_k^o = \epsilon_k^m + \epsilon_k^r \]

Measurement:
\[ E(\epsilon_k^m, \epsilon_k^{mT}) = M_k \]

Representation:
\[ E(\epsilon_k^r, \epsilon_k^{rT}) = R_k \]

Model uncertainty:
\[ E(\epsilon_k^q, \epsilon_k^{qT}) = Q_k \]

Forecast state using model
\[ x_{k+1}^f = \Psi_k x_k^a \]

Produce analysis using data
\[ x_k^a = x_k^f + K_k \left( m_k^o - H_k x_k^f \right) \]
\[ K_k = P_k^f H_k^T \left( H_k P_k^f H_k^T + R_k + Q_k \right)^{-1} \]

Update covariance using data
\[ P_k^a = P_k^f - K_k H_k P_k^f \]

Propagate covariance
\[ P_{k+1}^f = \Psi_k P_k^a \Psi_k^T + Q_k \]

Propagator must be a matrix!
Efforts I am Aware Of

- JPL/USC GAIM
- USU GAIM
- New effort: MEPS (Multi-model Ensemble Prediction System) at USU (PI), JPL and USC
- IDA4D – Astra Associates and APL (Gary Bust)
- EMPIRE (~IDA4D) – estimate drivers (Seebany Datta-Barua)
- TIEGCM+DART – ensemble Kalman filter
  - A monte-carlo approach to solving the Kalman filter equations
  - Tomoko Matsuo (NOAA)
- Reanalysis approach “global imaging”
  - Xinan Yue of UCAR
- New effort: Medium range thermosphere-ionosphere storm forecasts (Mannucci, PI) with USC and U Maryland, CCMC and collaborators
NEXT SLIDES ARE GOOD TO THINK ABOUT
(SPEED UP)
**Troposphere**

- Measured variables are temperature, pressure, water vapor, etc.
- Forecasts using first-principles equations (Navier-Stokes) are directly affected by measured quantities
  - Measurements provide information on momentum and energy
- Uncertainties arise due to “2nd order” microphysics
  - Clouds, soil processes, etc.

**Ionosphere**

- Measured variables are electron density
- Forecasts using first-principles equations (cold plasma) are directly affected by *unmeasured* “drivers”
  - Production and loss rates of electrons are determined by unmeasured quantities
- Uncertainties in forecast are “first-order”
  - Solar EUV spectrum
  - Neutral composition & winds
  - Electric fields
  - Particle precipitation
Estimate initial state at start of six-hour data assimilation cycle which yields forecast trajectory that best fits the data.
### Troposphere vs. Ionosphere

<table>
<thead>
<tr>
<th><strong>Troposphere</strong></th>
<th><strong>Ionosphere</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistent</td>
<td>Not as persistent</td>
</tr>
<tr>
<td>• Six-hour forecast steps</td>
<td>• 15-min. forecast steps</td>
</tr>
<tr>
<td>• Forecast is chaos limited</td>
<td></td>
</tr>
<tr>
<td>Millions of RT radiance measurements from multiple satellites</td>
<td>Smaller satellite datasets</td>
</tr>
<tr>
<td>• Date are decimated</td>
<td>• COSMIC GPS RO</td>
</tr>
<tr>
<td>• GPS RO provides new forecast skill</td>
<td>• UV nadir &amp; limb scans</td>
</tr>
<tr>
<td>Parameterized “drivers”</td>
<td>Parameterized drivers / couplings</td>
</tr>
<tr>
<td>• Cloud microphysics</td>
<td>• Solar EUV spectrum</td>
</tr>
<tr>
<td>Estimation Techniques</td>
<td>• Neutral composition &amp; winds</td>
</tr>
<tr>
<td>• 3DVAR, 4DVAR</td>
<td>• Electric fields</td>
</tr>
<tr>
<td>• Ensemble</td>
<td>• Particle precipitation</td>
</tr>
<tr>
<td></td>
<td>• M-I coupling</td>
</tr>
<tr>
<td></td>
<td>Estimation Techniques</td>
</tr>
<tr>
<td></td>
<td>• Leverage troposphere experience</td>
</tr>
</tbody>
</table>
Potential for Forecasting

- Troposphere: high potential
- Ionosphere: medium potential
  - Strongly driven during solar storms
- Magnetosphere: lower potential
  - Strongly driven
  - Sparse data

(Siscoe & Solomon, Space Weather, vol 4., 2006.)
• **Introduce JPL/USC GAIM**
  - First-principles physics model with an adjoint
  - Band-Limited Kalman Filter

• **Kalman Filter “Soup”**
  - Full, Reduced Rank, Correlation-Limited, Optimal Interpolation
  - Extended, Ensemble
JPL/USC GAIM First-Principles Model

- **3-D grid in a magnetic frame**

**Numerical Scheme**
- Finite volume on a fixed Eulerian grid
- Hybrid explicit-implicit time integration scheme

**Multiple ions:**
- O⁺, H⁺, He⁺

**Global and regional modeling**
by solving plasma hydrodynamic equations
Drivers of the Ionosphere: Coupling with Magnetosphere and Thermosphere

- Solar EUV radiation [F10.7 model]
  - Cause of the ionization
  - Solar flares

- Auroral particle precipitation
  - Cause of the ionization at high latitudes
  - Significant variations during storms and substorms

- Thermospheric composition & temperature [NRL MSIS]
  - Gas to be ionized
  - Loss of ionization due to chemical reactions [rate constants]
  - Global thermospheric circulation changes during storms

- Dynamics
  - Electric fields: originated from the magnetospheric and wind dynamo processes [Fejer-Schierless ExB drift model]
  - Thermospheric winds [HWM model]
  - Controlled by the geomagnetic field
  - Magnetospheric convection, penetration, and disturbance wind dynamo
Forward Model with an Adjoint

- **Driver Models**
  - NRL MSIS, HWM, Fejer-Scherliess ExB Drift (Fortran)

- **Eulerian Solver**
  - Variable-resolution, magnetic grid
  - Six ions: \( \text{O}^+, \text{H}^+, \text{He}^+, \text{N}_2^+, \text{O}_2^+, \text{NO}^+ \)

- **Computational Efficiency**
  - Adjoint computes all driver sensitivities in one pass
  - Assim. cycle: Run forward model once & adjoint just once
  - Multi-level physics cache:
    - Find in memory, pre-computed file, or generate

- **Optimized C++ (2\(^{nd}\) generation code)**
  - Object-oriented, Templated Matrix classes
  - High-performance numerics
  - Kudos to our C++ expert, Vardan Akopian
Kalman Filter Equations

State Model (propagator)

\[ x_{k+1}^f = \Psi_k x_k^f + \epsilon_{k}^q \]

Measurement Model

\[ m_k^o = H_k x_k^f + \epsilon_{k}^o \]

Noise Model

\[ \epsilon_{k}^o = \epsilon_{k}^m + \epsilon_{k}^r \]

Measurement:

\[ E(\epsilon_{k}^m, \epsilon_{k}^{mT}) = M_k \]

Representation:

\[ E(\epsilon_{k}^r, \epsilon_{k}^{rT}) = R_k \]

Model uncertainty:

\[ E(\epsilon_{k}^q, \epsilon_{k}^{qT}) = Q_k \]

Forecast state using model

\[ x_{k+1}^f = \Psi_k x_k^a \]

Produce analysis using data

\[ x_k^a = x_k^f + K_k \left( m_k^o - H_k x_k^f \right) \]

\[ K_k = P_k^f H_k^T \left( H_k P_k^f H_k^T + R_k + Q_k \right)^{-1} \]

Update covariance using data

\[ P_k^a = P_k^f - K_k H_k P_k^f \]

Propagate covariance

\[ P_{k+1}^f = \Psi_k P_k^a \Psi_k^T + Q_k \]
**Kalman Filter Equations**

**State Model (propagator)**

\[ x_{k+1}^f = \Psi_k x_k^f + \varepsilon_k^q \]

**Measurement Model**

(H = observation matrix)

\[ m_k^o = H_k x_k^f + \varepsilon_k^o \]

**Noise Model**

\[ \varepsilon_k^o = \varepsilon_k^m + \varepsilon_k^r \]

**Measurement:**

\[ \mathbb{E}(\varepsilon_k^m, \varepsilon_k^{mT}) = M_k \]

**Representation:**

\[ \mathbb{E}(\varepsilon_k^r, \varepsilon_k^{rT}) = R_k \]

**Model uncertainty:**

\[ \mathbb{E}(\varepsilon_k^q, \varepsilon_k^{qT}) = Q_k \]

**Forecast state using model**

\[ x_{k+1}^f = \Psi_k x_k^a \]

**Produce analysis using data**

\[ x_k^a = x_k^f + K_k (m_k^o - H_k x_k^f) \]

\[ K_k = P_k^f H_k^T (H_k P_k^f H_k^T + R_k + Q_k)^{-1} \]

**Update covariance using data**

\[ P_k^a = P_k^f - K_k H_k P_k^f \]

**Propagate covariance**

\[ P_{k+1}^f = \Psi_k P_k^a \Psi_k^T + Q_k \]
Kalman Filter Equations

State Model (propagator)
\[ x_{k+1}^f = \Psi_k x_k^f + \epsilon_k^q \]

Measurement Model
\[ m_k^o = H_k x_k^f + \epsilon_k^o \]

Noise Model
\[ \epsilon_k^o = \epsilon_k^m + \epsilon_k^r \]

Forecast state using model
\[ x_{k+1}^f = \Psi_k x_k^a \]

Produce analysis using data
\[ x_k^a = x_k^f + K_k (m_k^o - H_k x_k^f) \]
\[ K_k = P_k^f H_k^T (H_k P_k^f H_k^T + R_k + Q_k)^{-1} \]

Update covariance using data
\[ P_k^a = P_k^f - K_k H_k P_k^f \]

Propagate covariance
\[ P_{k+1}^f = \Psi_k P_k^a \Psi_k^T + Q_k \]

Propagator must be a matrix!
Setting the Covariance

• **P = Covariance matrix**
  - Iono. Variability & Correlations drawn from model runs
  - Diagonal: Variability in density for that grid cell
  - Off-diagonal: Use physical iono. correlation lengths

• **Q “bump”**
  - Handles model uncertainty
  - Increase density errors (diagonal) at every Kalman update
  - Otherwise filter assumes prior model is perfect

• **Balance between model and data**
  - Q versus quantity of TEC data with ~2 TECU noise
  - For COSMIC TEC links, can drive profile “hard” with the data
SPEED UP
Kalman Filter Characteristics

- **Linear vs. non-linear**
  - Non-linear formulation usually requires iteration
  - Full covariance matrix is huge (100,000 by 100,000 and up)

- **Physics Model: Black Box or Integrated**
  - Black Box:
    - Estimate differences from background model
    - Differences treated as a stochastic parameter (Gauss-Markov)
  - Integrated into the Kalman filter:
    - Have propagation matrix so can propagate covariance

- **State vector**
  - Density state only, or log(density)
  - Augmented with (non-linear) driver parameters

- **Approach to Covariance**
  - Full or sparse; fixed, propagated, directly computed from ensemble
The World of Kalman Filters

- **Full Kalman-Bucy filter**
  - Optimal, but not tractable for high 3D resolutions
  - Full covariance matrix is huge (100,000 by 100,000 and up)

- **Reduced Rank Kalman filter**
  - Run physics model on 1M cells, but filter on 30,000
  - Example: USU Gauss-Markov GAIM

- **Optimal Interpolation**
  - Fixed covariance matrix, don’t propagate covariance

- **Limited-Correlation Kalman filter (band-limited)**
  - Covariance matrix is sparse, covariance is propagated
  - Off-diagonal covariance beyond physical correlation lengths in the medium is discarded.
  - Implementation harder, still computationally intensive
## Optimal Interpolation

### State Model (propagator)

$$ x_{k+1}^f = \Psi_k x_k^f + \epsilon_k^q $$

### Measurement Model

(H = observation matrix)

$$ m_k^o = H_k x_k^f + \epsilon_k^o $$

### Noise Model

$$ \epsilon_k^o = \epsilon_k^m + \epsilon_k^r $$

### Measurement:

$$ E(\epsilon_k^m, \epsilon_k^m^T) = M_k $$

### Representation:

$$ E(\epsilon_k^r, \epsilon_k^r^T) = R_k $$

### Model uncertainty:

$$ E(\epsilon_k^q, \epsilon_k^q^T) = Q_k $$

### Forecast state using model

$$ x_{k+1}^f = \Psi_k x_k^a $$

### Produce analysis using data

$$ x_k^a = x_k^f + K_k (m_k^o - H_k x_k^f) $$

$$ K_k = P_k^f H_k^T (H_k P_k^f H_k^T + R_k + Q_k)^{-1} $$

### Update covariance using data

$$ P_k^a = P_k^f - K_k H_k P_k^f $$

### Propagate covariance

$$ P_{k+1}^f = \Psi_k P_k^a \Psi_k^T + Q_k $$

*P = fixed covariance matrix*
### Band-Limited Kalman Filter

**State Model (propagator)**

\[ x_{k+1}^f = \Psi_k x_k^a + \varepsilon_k^q \]

**Measurement Model**

(H = observation matrix)

\[ m_k^o = H_k x_k^f + \varepsilon_k^o \]

**Noise Model**

\[ \varepsilon_k^o = \varepsilon_k^m + \varepsilon_k^r \]

**Measurement:**

\[ \mathbb{E}(\varepsilon_k^m, \varepsilon_k^{mT}) = M_k \]

**Representation:**

\[ \mathbb{E}(\varepsilon_k^r, \varepsilon_k^{rT}) = R_k \]

**Model uncertainty:**

\[ \mathbb{E}(\varepsilon_k^q, \varepsilon_k^{qT}) = Q_k \]

**Forecast state using model**

\[ x_{k+1}^f = \Psi_k x_k^a \]

**Produce analysis using data**

\[ x_k^a = x_k^f + K_k (m_k^o - H_k x_k^f) \]

\[ K_k = P_k^f H_k^T (H_k P_k^f H_k^T + R_k + Q_k)^{-1} \]

**Update covariance using data**

\[ P_k^a = P_k^f - K_k H_k P_k^f \]

**Propagate covariance**

\[ P_{k+1}^f = \Psi_k P_k^a \Psi_k^T + Q_k \]

*P is a sparse matrix*

*Operation count scales with N rather than N^2*
Band-Limited Kalman Filter

- Approximate Kalman: Save only part of covariance matrix based on physical correlation lengths.

- Tested extensively with real data: Ground GPS TEC from 100-200 global sites.

- Validate densities against:
  - Vertical TEC obs. From TOPEX
  - Ionosonde FoF2, HmF2, & bottomside profiles
  - Slant TEC obs. from independent ground GPS sites.
  - Density profiles retrieved from space-based GPS occultations
## Summary Of No. Of Operations

<table>
<thead>
<tr>
<th>Approach</th>
<th>No. of Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Kalman</td>
<td>$28800 \times N^2 + M \times N^2$</td>
</tr>
<tr>
<td>Optimal Interpolation</td>
<td>$2 \times N \times M$</td>
</tr>
<tr>
<td>Band Limited</td>
<td>$A \times N \times M$</td>
</tr>
</tbody>
</table>

- $M = \text{No. of measurements}$
- $N = \text{No. of Voxels}$
- $A = \text{No. of neighbor elements with non-zero covariance}$
No. of Operations
Per 100,000 TEC Measurements
Reduced Rank, Difference Filter

**Properties**
- Example: USU Gauss-Markov GAIM
- Run physics model on high-resolution magnetic grid (1 million cells)
- Assimilate obs. on low-resolution geographic grid (10,000 cells x 3)
- Estimate difference from “black box” physics model

**Advantages**
- Can use physics model unchanged, computationally tractable

**Disadvantages**
- Data update occurs only on low-res. grid, high-res. physics wasted
- Requires two-way (potentially non-linear) interpolation between high and low-res. grids
- Resolution limited, potential loss of fidelity
- As usual, covariance determines balance between model and data.
Limited-Correlation Kalman Filter

• Properties
  • Example: JPL/USC Band-Limited Kalman filter
  • Physics model and filter use same magnetic, variable-res. grid (150,000 cells)
  • Forward model computes propagator matrix so can propagate covariance matrix
  • Covariance matrix is sparse, correlations beyond physical lengths discarded

• Advantages
  • Can compute all of the Kalman filter equations
  • Computationally tractable, answer approximates true state

• Disadvantages
  • Physics model specially developed to support Kalman filter
  • Solver and filter use same or commensurate magnetic grids
  • Resolution somewhat limited due to computational cost
SLOW DOWN HERE
Ensemble Kalman Filter

<table>
<thead>
<tr>
<th>State Model (propagator)</th>
<th>( x^f_{k+1} = \Psi_k x^f_k + \varepsilon^q_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement Model</td>
<td>( m^o_k = H_k x^f_k + \varepsilon^o_k )</td>
</tr>
<tr>
<td>Noise Model</td>
<td>( \varepsilon^o_k = \varepsilon^m_k + \varepsilon^r_k )</td>
</tr>
<tr>
<td>Measurement:</td>
<td>( E(\varepsilon^m_k, \varepsilon^{m^T}_k) = M_k )</td>
</tr>
<tr>
<td>Representation:</td>
<td>( E(\varepsilon^r_k, \varepsilon^{r^T}_k) = R_k )</td>
</tr>
<tr>
<td>Model uncertainty:</td>
<td>( E(\varepsilon^q_k, \varepsilon^{q^T}_k) = Q_k )</td>
</tr>
</tbody>
</table>

Forecast state using model
\[ x^a_{k+1} = \Psi_k x^a_k \]

Produce analysis using data
\[ x^a_k = x^f_k + K_k \left( m^o_k - H_k x^f_k \right) \]

Update covariance using data
\[ K_k = P^f_k H_k^T \left( H_k P^f_k H_k^T + R_k + Q_k \right)^{-1} \]

Propagate covariance
\[ P^a_k = P^f_k - K_k H_k P^f_k \]

P directly computed from ensemble.
The World of Kalman Filters (2)

- **Ensemble Kalman filter**
  - Forget covariance update & propagation equations
  - Monte-Carlo implementation of Bayesian update problem
  - Run “ensemble” of state vectors
  - Replace covariance matrix (P) with sample covariance computed from ensemble
  - Useful for state augmented with drivers
  - **Challenge:** Does ensemble properly sample iono. variability?
  - Computationally efficient if ensemble size is small (20-40)
  - Easily parallelizable

- **Extended Kalman filters (add drivers)**
  - Augment system state with estimated driver parameters
  - Any type of Kalman can be “extended”
  - But watch out for non-linearity
Kalman Assimilation Runs: Case Studies

- **Three runs:**
  - GAIM Climate (no data)
  - Ground GPS TEC (200 sites)
  - Ground + COSMIC links (upward & occultation)

- **Medium Resolution runs:**
  - 2.5 Lat. in Deg.
  - 10.0 Lon. in Deg.
  - 40.0 Alt. in km

- **100,000 voxels**

- **Sparse Kalman filter:**
  - Update & propagate covariance
  - Truncate off-diagonal covariance that is beyond physical correlation lengths

Intersections of:
- magnetic field lines,
- magnetic geopotential lines
- and magnetic longitudes
Validating a Kalman Filter

- **Pre-fit Residuals**
  - How close is the physics model (climate) to today’s weather?
- **Watch your covariance (errors & correlations)**
  - A priori covariance matches iono. correlation lengths
  - Model uncertainty: Q-bump reasonable?
- **Post-fit Residuals**
  - Did the filter fit the data? Were outliers rejected?
  - Inconsistent datatypes: Is one dataset inconsistent?
- **Independent Accuracy Validation (continuously)**
  - TEC space: Global morphology? Match JASON VTEC?
  - Profile space: Match ISR profiles, ionosonde FoF2/HmF2
  - Compare to Abel-inversion profiles (proxy for truth)
  - If validation data unavailable, withhold input data
- **Improve model, tune covariance, and repeat.**
SPEED UP. LOTS OF VALIDATION RESULTS
Illustration for TEC data, GAIM Prefit and Postfit Residuals
GAIM versus Abel Profiles

profile static UT: 20060626_181200 UT: 15.4 ~Lat: 029 ~Lon: 313

profile static UT: 20060626_183200 UT: 15.4 ~Lat: 029 ~Lon: 313

profile static UT: 20060626_182000 UT: 15.3 ~Lat: 031 ~Lon: 315
Comparing GAIM NmF2 to Abel:
June 26, 2006

Climate (no data)

Ground only

Ground + Space
NmF2 Comparison: Bear Lake
Jicamarca ISR Study for Sept 21, 2006

1. UT 15:36

2. UT 15:48

3. UT 16:36

COSMIC
UT 15:30

1. 2. 3.
Statistical Summary: RMS of Jason 2 VTEC Differences, Nov. 21, 2008

Day-Time RMS VTEC Differences Compared to Jason-2 on Nov 21, 2008

Night-Time RMS VTEC Differences Compared to Jason-2 on Nov 21, 2008
Summary of Results (2)

HmF2 Comparison of Low and Medium Resolution Runs for Sept 21, 2006

- GO = Ground-GPS only
- GD = Ground + down-looking COSMIC
GAIM Validation Using Jason-2 Vertical TEC for June 26

Ground-data only

Ground and space data

COSMIC
GAIM Driven By Ground GPS Only versus JASON VTEC

June – Nov. 2004: 137 days
SLOW DOWN – Driver Estimation Using JPL/USC GAIM
Global Assimilative Ionospheric Model
Data Assimilation Process

Driving Forces → Physics Model → Mapping State To Measurements

Adjustment Of Parameters → 4DVAR → Kalman Filter

State and covariance Forecast → State and covariance Analysis → Innovation Vector
Global Assimilative Ionospheric Model
Data Assimilation Process

- Kalman Filter
  - Recursive Filtering
  - Covariance estimation and state correction
    - Optimal interpolation
    - Band-Limited Kalman filter
Global Assimilative Ionospheric Model
Data Assimilation Process

- Driving Forces
  - Adjustment Of Parameters

- Physics Model
  - State and covariance Forecast
  - State and covariance Analysis

- Mapping State To Measurements
  - Innovation Vector

- 4DVAR

- Kalman Filter
  - Recursive Filtering
  - Covariance estimation and state correction
  - Optimal interpolation
  - Band-Limited Kalman filter

- 4-Dimensional Variational Approach
  - Minimization of cost function by estimating driving parameters
  - Non-linear least-square minimization
  - Adjoint method to efficiently compute the gradient of cost function
  - Parameterization of model “drivers”
Estimation of Ionospheric Dynamical Drivers

- Observation System Simulation Experiments (OSSE) to estimate "perturbed" drivers at low latitudes:
  - Production terms
  - $E \times B$ vertical drift velocity
  - Neutral winds
- Synthetic ground GPS TEC data
Need Sensitivities to Adjust Drivers

- Use Physics Model as a Black Box
  - Run Forward Model twice for different parameter choices
  - Difference resulting densities to compute numerical derivatives (grid of sensitivities)
  - Repeat for each driver parameter
  - Implies large number of forward model runs

- Develop Adjoint Model corresponding to the Forward Model
  - Run Forward Model once
  - Run Adjoint Model once
  - During Adjoint run all parameter sensitivities are computed
  - Use sensitivities in 4DVAR variational optimization or extended Kalman filter to adjust drivers
Improved Drivers => Improved Forecasting

- Improved drivers enable more accurate “nowcast” and forecast of 3D electron density.

- Plot differences between simulated ionospheric “weather” and assimilation results for vertical TEC and Ne profiles.
Optimization Approach: 4DVAR

\[ J(n; \alpha) = \sum_{k=1}^{m} \left\| y_k - H_k n(t_k; \alpha) \right\|^2 + \beta \left\| n - n_0 \right\|^2 + \lambda \left\| \alpha - \alpha_0 \right\|^2 \]

\[ \nabla J(\dot{\alpha}) = 0 \]

\[ v_{eq}(t) = v_{eq,0}(t) + \sum_{k=1}^{N} \alpha_k \phi_k(t) \]

\[ F(r) = F_0(r) + \frac{1}{7} \sum_{i=1}^{7} w_i(\rho', \sigma) f_i(r') \]

- Non-linear least squares minimization
  - **Cost function** model deviation from observations
  - **Adjoint** gradient of cost function: computational efficiency
  - **Minimization**: using Newton’s method by estimating driving parameters
  - **Parameterization** of model drivers

Estimate ionospheric drivers to optimize state
Optimization Approach: 4DVAR

\[ J(n; \alpha) = \sum_{k=1}^{m} \left\| y_k - H_k n(t_k; \alpha) \right\|^2 + \beta \left\| n - n_0 \right\|^2 + \lambda \left\| \alpha - \alpha_0 \right\|^2 \]

- **Non-linear least squares minimization**
- **Cost function** model deviation from observations
- **Adjoint** gradient of cost function: computational efficiency
- **Minimization**: using Newton’s method by estimating driving parameters
- **Parameterization** of model drivers

\[ \nabla J(\dot{\alpha}) = 0 \]

\[ \nu_{eq}(t) = \nu_{eq,0}(t) + \sum_{k=1}^{N} \alpha_k \phi_k(t) \]

\[ F(r) = F_0(r) + \frac{1}{7} \sum_{i=1}^{7} w_i(\rho', \sigma) f_i(r') \]

Estimate ionospheric drivers to optimize state
Optimization Approach: 4DVAR

\[ J(n; \alpha) = \sum_{k=1}^{m} \left\| y_k - H_k n(t_k; \alpha) \right\|^2 + \beta \left\| n - n_0 \right\|^2 + \lambda \left\| \alpha - \alpha_0 \right\|^2 \]

\[ \nabla J(\dot{\alpha}) = 0 \]

\[ v_{eq}(t) = v_{eq,0}(t) + \sum_{k=1}^{N} \alpha_k \phi_k(t) \]

\[ F(r) = F_0(r) + \frac{1}{7} \sum_{i=1}^{7} w_i(\rho', \sigma) f_i(r') \]

- Non-linear least squares minimization
  - \textit{Cost function} model deviation from observations
  - \textit{Adjoint} gradient of cost function: computational efficiency
  - \textit{Minimization}: using Newton’s method by estimating driving parameters
  - \textit{Parameterization} of model drivers

Estimate ionospheric drivers to optimize state
Optimization Approach: 4DVAR

\[ J(n; \alpha) = \sum_{k=1}^{m} \left\| y_k - H_k n(t_k; \alpha) \right\|^2 + \beta \left\| n - n_0 \right\|^2 + \lambda \left\| \alpha - \alpha_0 \right\|^2 \]

\[ \nabla J(\dot{\alpha}) = 0 \]

\[ v_{eq}(t) = v_{eq,0}(t) + \sum_{k=1}^{N} \alpha_k \phi_k(t) \]

\[ F(r) = F_0(r) + \frac{1}{7} \sum_{i=1}^{7} w_i(\rho', \sigma) f_i(r') \]

- Non-linear least squares minimization
  - **Cost function** model deviation from observations
  - **Adjoint** gradient of cost function: computational efficiency
  - **Minimization**: using Newton’s method by estimating driving parameters
  - **Parameterization** of model drivers

Estimate ionospheric drivers to optimize state
Optimization Approach: 4DVAR

\[ J(n; \alpha) = \sum_{k=1}^{m} \left\| y_k - H_k n(t_k; \alpha) \right\|^2 + \beta \left\| n - n_0 \right\|^2 + \lambda \left\| \alpha - \alpha_0 \right\|^2 \]

\[ \nabla J(\dot{\alpha}) \]

\[ \nabla J(\dot{\alpha}) = 0 \]

\[ \nu_{eq}(t) = \nu_{eq,0}(t) + \sum_{k=1}^{N} \alpha_k \phi_k(t) \]

\[ F(r) = F_0(r) + \frac{1}{7} \sum_{i=1}^{7} w_i(\rho', \sigma) f_i(r') \sum_{i=1}^{7} w_i \]

- Non-linear least squares minimization
  - Cost function model deviation from observations
  - Adjoint gradient of cost function: computational efficiency
  - Minimization: using Newton’s method by estimating driving parameters
  - Parameterization of model drivers

Estimate ionospheric drivers to optimize state
Optimization Approach: 4DVAR

\[ J(n; \alpha) = \sum_{k=1}^{m} \left\| y_k - H_k n(t_k; \alpha) \right\|^2 + \beta \| n - n_0 \|^2 + \lambda \| \alpha - \alpha_0 \|^2 \]

\[ \nabla J(\dot{\alpha}) \]

\[ \frac{1}{\nabla J(\dot{\alpha})} = 0 \]

\[ v_{eq}(t) = v_{eq,0}(t) + \sum_{k=1}^{N} \alpha_k \phi_k(t) \]

\[ F(r) = F_0(r) + \frac{1}{7} \sum_{i=1}^{7} w_i(\rho', \sigma) f_i(r') \]

- Non-linear least squares minimization
  - Cost function: model deviation from observations
  - Adjoint: gradient of cost function: computational efficiency
- Minimization: using Newton’s method by estimating driving parameters
- Parameterization of model drivers

Estimate ionospheric drivers to optimize state
Optimization Approach: 4DVAR

\[ J(n; \alpha) = \sum_{k=1}^{m} \left\| y_k - H_k n(t_k; \alpha) \right\|^2 + \beta \left\| n - n_0 \right\|^2 + \lambda \left\| \alpha - \alpha_0 \right\|^2 \]

\[ \nabla J(\dot{\alpha}) = 0 \]

\[ v_{eq}(t) = v_{eq,0}(t) + \sum_{k=1}^{N} \alpha_k \phi_k(t) \]

\[ F(r) = F_0(r) + \frac{1}{7} \sum_{i=1}^{7} w_i(\rho', \sigma) f_i(r') \]

- **Non-linear least squares minimization**
  - Cost function model deviation from observations
  - **Adjoint** gradient of cost function: computational efficiency
  - **Minimization:** using Newton’s method by estimating driving parameters
  - **Parameterization** of model drivers

Estimate ionospheric drivers to optimize state
SPEED UP
Two Things I Do Not Understand

- The adjoint algorithm
- Public key encryption
Driver Grids and GPS Stations

- Number of stations: 12/07/2002: 31
- Observation links: ~2240/ hour, sampled at 5-minute epochs
Vertical Drift Velocity

- ExB drift at geomag. equator versus Local Time
  - 9 non-uniform cubic splines
  - N uniform cubic splines (8 to 24)
Concluding
Efforts I am Aware Of

- JPL/USC GAIM
- USU GAIM
- New effort: MEPS (Multi-model Ensemble Prediction System) at USU (PI), JPL and USC
- IDA4D – Astra Associates and APL (Gary Bust)
- EMPIRE (~IDA4D) – estimate drivers (Seebany Datta-Barua)
- TIEGCM+DART – ensemble Kalman filter
  - A monte-carlo approach to solving the Kalman filter equations
  - Tomoko Matsuo (NOAA)
- Reanalysis approach “global imaging”
  - Xinan Yue of UCAR
- New effort: Medium range thermosphere-ionosphere storm forecasts (Mannucci, PI) with USC and U Maryland, CCMC and collaborators
- DREAM (radiation belt) – Los Alamos National Labs
Summary

• Figure out what you want to do and what data assimilation will do for you
  • What is the role of deterministic dynamics in what you want to do?
• What are the scientific benefits?
  • Provide improved “specification” for understanding the coupled system
  • Understand cause and effect
  • Understand your assumptions and errors
• Learn what has been done in this and other fields
• Have a strong validation system set up and keep checking
  • Recall: data assimilation is not a “natural” process