Data Assimilation:

One Perspective, Some Examples

Gary Egbert, Oregon State University

- Some basics, examples (from oceanography)
- A geodynamic example: short term tectonics

Bottom line: going forward CIG should consider development of capabilities for data assimilation (and inversion or “imaging”) … if we plan ahead it will be a lot easier

Data Assimilation: Estimate State (Ocean, Atmosphere, Earth) Using:

(1) Dynamical equations $\mathcal{S}\mathbf{u} = \mathbf{f}$

$\mathbf{u}$ "Ocean" state

$\mathbf{f}$ Forcing, initial conditions, boundary conditions

(2) Data $\mathbf{d} = \mathcal{L}\mathbf{u}$

$\mathbf{d}$ Vector of observations

$\mathcal{L}$ Data functionals

Over-determined problem (more constraints than unknowns)

***Choose* u to compromise between, data & dynamics**

Numerous applications, approaches to implementation

Example

Assimilation of surface elevation, velocity in a shallow-water model of alongshore currents in the surf zone, turbulent regime
(synthetic data; Kurapov et al., 2007)

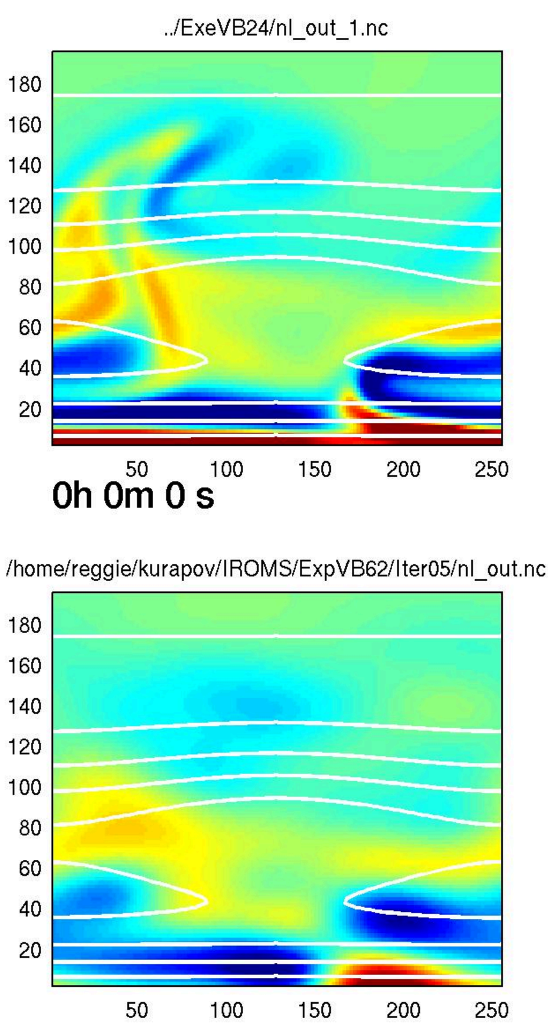
One perspective on DA:

→ keep a dynamical model "on track"

… weather forecasting

Vorticity

True solution:

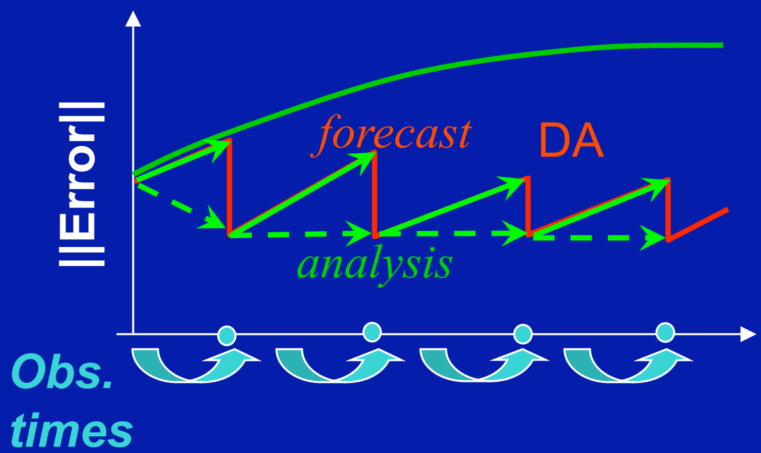


NL solution forced by the estimated forcing

Kalman Filter: a ***sequential*** method

Time dependent dynamics:

$$\left[\,\partial_t - S_0\,\right]\mathbf{u} = \mathbf{n} \longleftarrow \textit{White noise}$$



Forecast error covariance:

$$\mathbf{K} = E\left[(\mathbf{u}_{\text{forecast}} - \mathbf{u}_{\text{true}})(\mathbf{u}_{\text{forecast}} - \mathbf{u}_{\text{true}})^T\right]$$

Optimal correction to forecast:

$$\mathbf{u}_{\text{anal.}} = \mathbf{u}_{\text{fc}} + \mathbf{G}(\mathbf{d} - L\mathbf{u}_{\text{fc}}) \qquad \mathbf{G} = \mathbf{K}\mathbf{H}^T\left(L\mathbf{K}H^T + \mathbf{C}_{\text{d}}\right)^{-1}$$

In general, K is time dependent … and impractical to calculate for even modest size state space

Sequential data assimilation in practice:

- Reduced state space Kalman filter: try to describe essential dynamics in a *very* reduced state space

- Ensemble Kalman filter: approximate time evolution of the forecast error covariance with an ensemble of assimillation experiments

- Assume K is constant in time, estimate somehow using some sort of ensemble calculation

- Take a wild guess at K … something simple and not too hard to work with

A different perspective:

***b*ring data and dynamical model into a common framework to**

→ optimally interpolate observations, incorporating dynamics/physical constraints

→ estimate inputs (forcing, initial/ boundary conditions)

→Discover "missing model physics"; test hypotheses

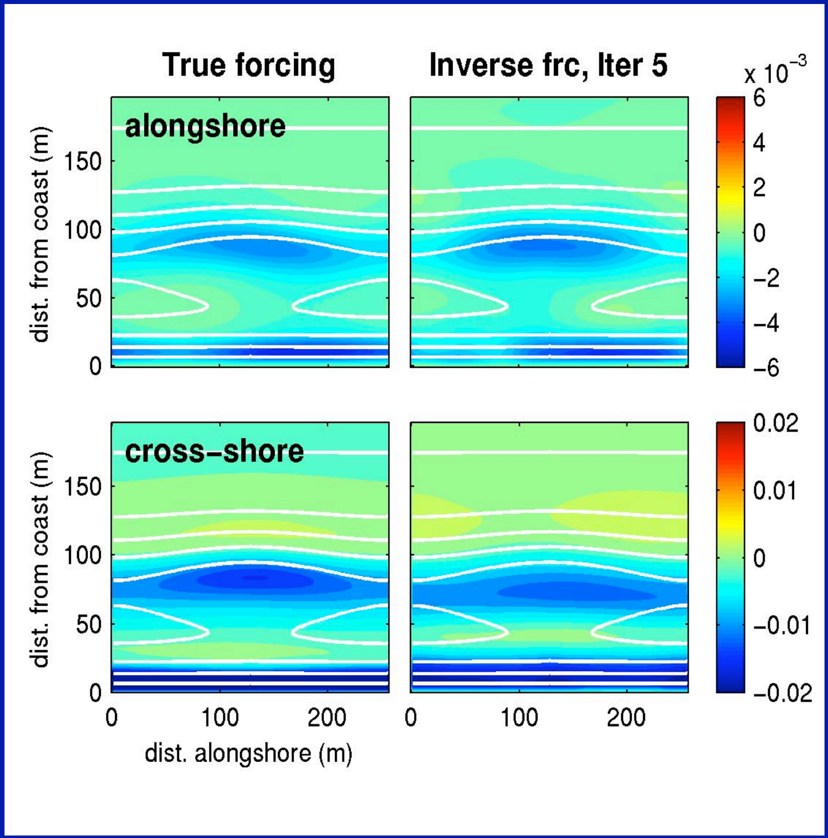
(→ estimate model parameters)

More sensible for (most) geodynamic applications?

SWE Example:

- **True solution: unsteady, irregular flow in response to steady forcing**

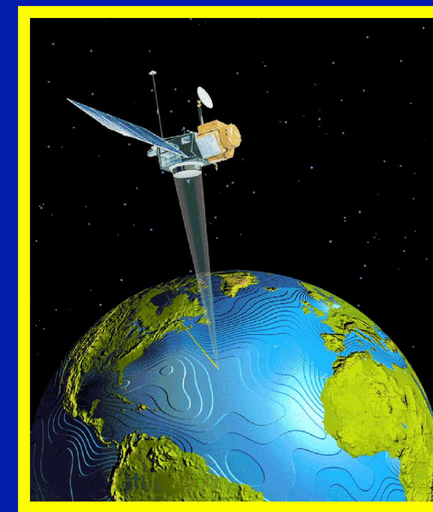
- **DA: corrects initial conditions and forcing**

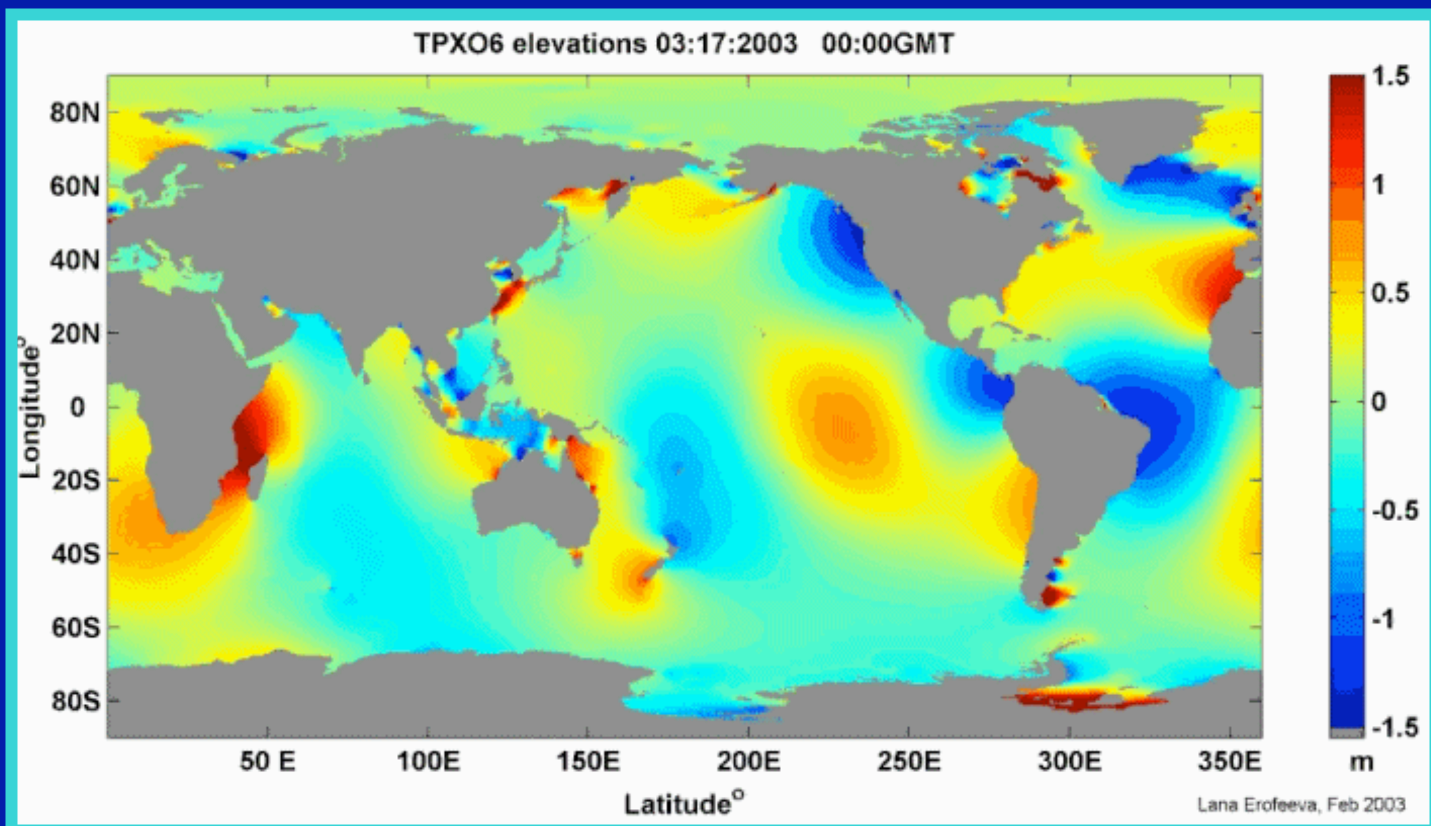


Example: global ocean tide *(Egbert et al. 1994)*

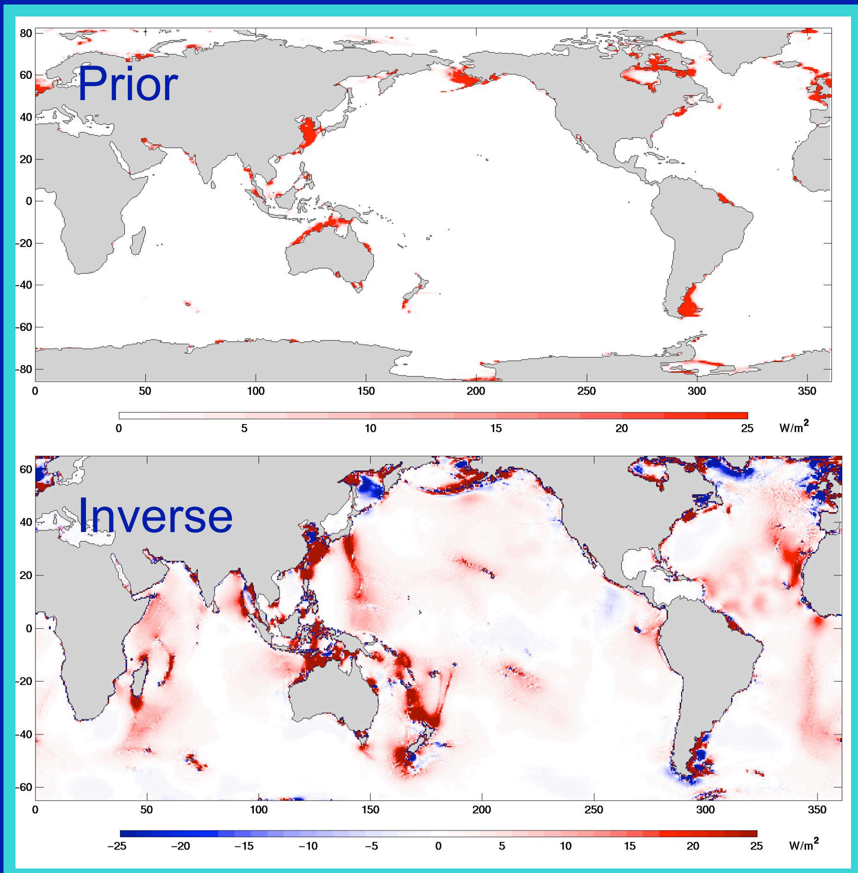
Dynamics: SWE, 8 tidal constituents (10^6 elements in state space)

Data: TOPEX/Poseidon altimetry (10^6 independent data)





Tidal energy dissipation

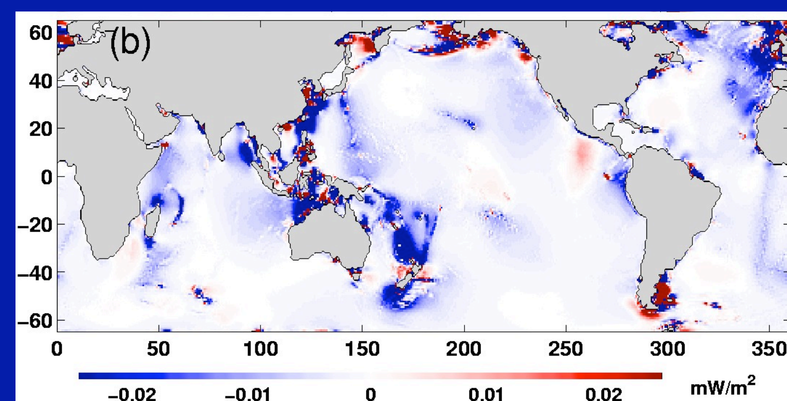


Dissipation is enhanced over rough topography in the deep ocean energy is extracted from surface tide, converted to internal waves

Dynamical equations are not satisfied exactly:

$$S\hat{u} - \mathbf{f} = \mathbf{r}$$

residuals contain information about processes that are omitted or modeled poorly



Work done by dynamical residuals

Variational Data Assimilation

Estimate state (e.g., deformation) combining :

Dynamical equations	$Su = f + \delta f$	Allow for errors in dynamical equations (forcing, boundary and initial conditions, missing physics) and data
Data	$d = \mathcal{L}u + \varepsilon$	

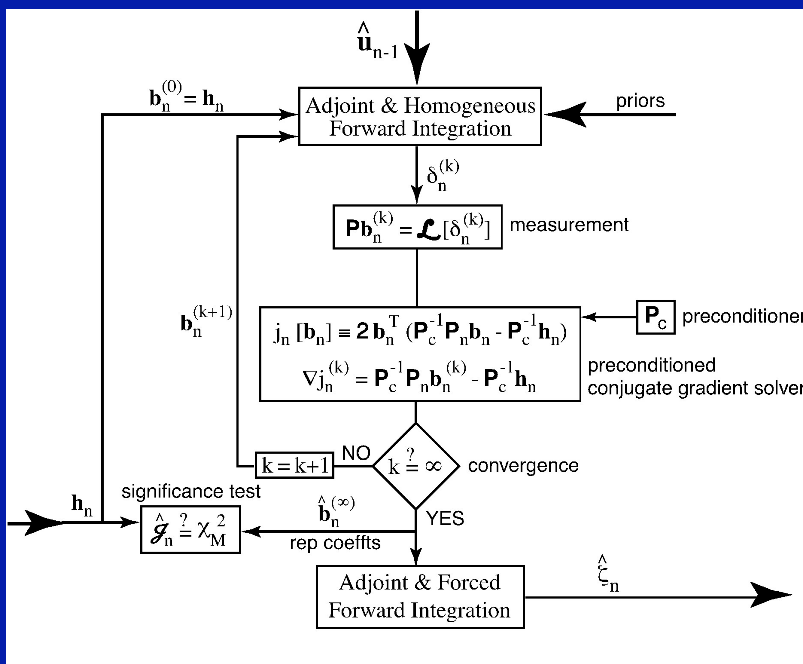
Minimize penalty functional

$$\mathcal{J}[u] = (d - \mathcal{L}u)^{\dagger}\,\Sigma_d^{-1}\,(d - \mathcal{L}u) + (Su - f)^{\dagger}\,\Sigma_f^{-1}\,(Su - f)$$

Error covariances encode a priori beliefs about magnitude, spatial/temporal correlation structure of errors in forcing, boundary and initial conditions, data

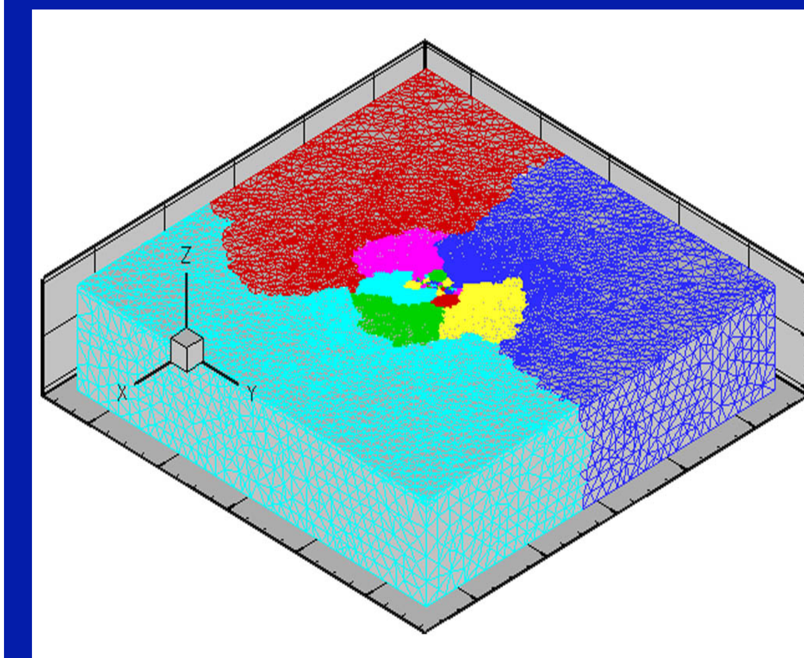
Application of variational data assimilation to a geodynamic problem

Inverse Ocean Modeling (IOM) : an NSF ITR project focused on ocean applications (A. Bennett)



make complex data assimilation algorithms accessible to a wider audience of modelers

GeoFEST: finite element visco-elastic modeling code (G. Lyzenga et al.)



Developed as a client application of the IOM

Euler-Lagrange Equations

(necessary conditions for a minimum $\mathcal{J}$)

Backward: $$S_{TL}^{\dagger}\lambda = \boldsymbol{L}^{\dagger}\Sigma_{\mathrm{d}}^{-1}(\mathrm{d} - \boldsymbol{L}\mathrm{u})$$

Forward: $$\mathrm{S}\,\mathrm{u} \rlap{=}\lambda\, \mathrm{f} + \Sigma_f$$

(Non-linear) coupled system for state variable **u**
and adjoint variable λ

S_{TL} is the tangent linear at the solution **u**

$S_{TL}^{\dagger}$ is the adjoint of the tangent linear

IOM solves linearized E-L equations, iterating to allow for non-linearity

Direct Representer Approach (Linear)

Can show: $\hat{\mathbf{u}} = \mathbf{u}_0 + \sum_k \beta_k \mathbf{r}_k$ ⟵ **representers:** obtained by solving

prior

$$\mathbf{S}_{TL}^{\dagger}\boldsymbol{\lambda}_k = L_k \qquad \mathbf{S}_{TL}\mathbf{r}_k = \Sigma_{\mathbf{f}}\boldsymbol{\lambda}_k$$

Coefficients b satisfy: $(\mathbf{R} + \Sigma_{\mathbf{d}})\boldsymbol{\beta} = \mathbf{d} - \mathcal{L}\mathbf{u}$

representer matrix $R_{jk} = \mathcal{L}_j \mathbf{r}_k$

Solve adjoint, forward problem for each observation

IOM uses an “indirect representer” algorithm: solve for b using conjugate gradients, without forming the matrix R

GeoFEST Forward Model Equations

Equation for evolution of displacement field

Stiffness matrix

visco-plastic strain relaxation

$$\mathbf{K}\frac{d\mathbf{u}}{dt}\boldsymbol{\beta} - \mathbf{B}^{\dagger}\mathbf{D}\;\mathbf{f}(\quad) =$$

forcing

elastic constants

Equation for evolution of stress tensor

$$\frac{d\boldsymbol{\sigma}}{dt} - \mathbf{D}\left[\boldsymbol{\beta}\,\boldsymbol{\sigma}\frac{d\mathbf{u}}{dt} - (\quad)\right] = 0$$

Finite element solver uses semi-implicit time-stepping scheme

(based on linearization $\boldsymbol{\beta}(\boldsymbol{\sigma}_0 + \delta\boldsymbol{\sigma}) \approx \boldsymbol{\beta}(\boldsymbol{\sigma}_0) + \boldsymbol{\beta}'\delta\boldsymbol{\sigma}$)

Developing Tangent Linear and Adjoint for GeoFEST

- **Based on existing code (but not a line-by-line adjoint!)**
- **Divide and concur: e.g,**

  → strip off complications in the way GeoFEST forcing has been implemented

  → work out adjoint for time stepping scheme in terms of adjoints of spatial operators

  → develop adjoints of spatial operators as needed

- **In the end, very little new code actually needed!**

Discrete Equations

Tangent Linear $\boldsymbol{\lambda}_0, \delta\boldsymbol{\lambda}_n, \boldsymbol{\varepsilon}_n \longrightarrow \mathbf{u}_0, \delta\mathbf{u}_n, \ _n$

$\longleftarrow \mathbf{u}_0, \delta\mathbf{u}_n, \ _n$ **Adjoint**

$\boldsymbol{\lambda}_0, \delta\boldsymbol{\lambda}_n, \boldsymbol{\varepsilon}_n$

$$\mathbf{K}_0\mathbf{u}_0\boldsymbol{\lambda}_0 = \ _0$$

$$\boldsymbol{\sigma}_0 - \mathbf{DBu}_0 = \boldsymbol{\varepsilon}_0$$

For n = 0, N

$$\mathbf{K}_{n+1}\delta\mathbf{u}_{n+1} - \mathbf{B}^{\dagger}\tilde{\mathbf{D}}\tilde{\boldsymbol{\beta}}'_{n+1}\ \boldsymbol{\sigma}_{n+1}\ _n = \boldsymbol{\lambda}\delta\ _{n+1}$$

$$\boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n - \delta t\tilde{\mathbf{D}}_{n+1}[\mathbf{B}\delta\mathbf{u}_{n+1} - \tilde{\boldsymbol{\beta}}'_{n+1}\boldsymbol{\sigma}_n] = \boldsymbol{\varepsilon}_{n+1}$$

$$\boldsymbol{\varepsilon}_0 = \boldsymbol{\sigma}_0$$

For n = N,N-1,…1

$$\mathbf{K}\boldsymbol{\lambda}_n\ _n - \delta t\mathbf{B}^{\dagger}\tilde{\mathbf{D}}\boldsymbol{\varepsilon}_n\ _n = \mathbf{u}\delta\ _n$$

$$\boldsymbol{\varepsilon}_{n-1} - \boldsymbol{\varepsilon}_n - \tilde{\boldsymbol{\beta}}'_{n-1}\tilde{\mathbf{D}}_n[\mathbf{B}\boldsymbol{\lambda}_n - \delta t\boldsymbol{\varepsilon}_n] = \boldsymbol{\sigma}_{n-1}$$

$$\mathbf{K}\boldsymbol{\lambda}_0 - \mathbf{B}^{\dagger}\mathbf{D}\boldsymbol{\varepsilon}_0 = \mathbf{u}_0$$

Virtually everything needed for TL and ADJ were already coded for forward solver

- Linearization of rheology → already coded for implicit time-step scheme
- Adjoint of operator B (gradients of displacement, to compute stress) is already coded (divergence of stress, to compute force balance
- Constitutive operators (elastic mapping D, linearized rheology β') are symmetric (self adjoint)

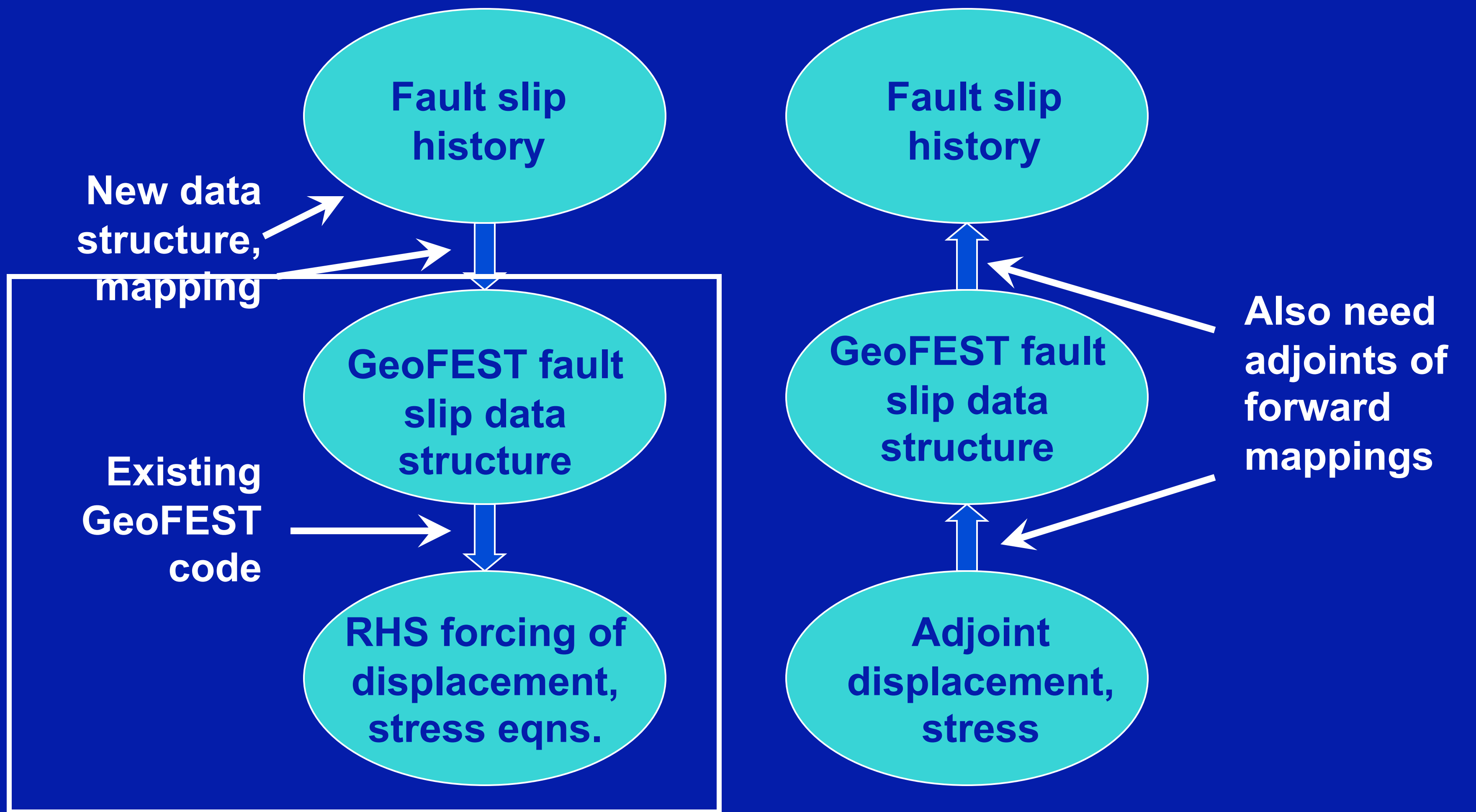
Comparatively minor reorganization of code required

GeoFEST Forcing

- specified displacement of boundary
- specified stress on boundary
- specified fault displacement, at specific times
  implemented through "split nodes"

Focus on the last … allows use of GeoFEST for estimation of fault slip history (e.g., could be applied to mapping spatio-temporal structure of ETS)

Estimation of fault slip history …



Relation to “parameter estimation” (“inversion”)

Unknown elastic parameters: $\mathbf{D} = \mathrm{diag}[\nu(\theta)]$

Data misfit:

$$\mathcal{J}[\theta] = \left(\mathbf{d} - \boldsymbol{L}\boldsymbol{S}_{\theta}^{-1}\mathbf{f}\right)^{T} \Sigma_{\mathrm{d}}^{-1} \left(\mathbf{d} - \boldsymbol{L}\boldsymbol{S}_{\theta}^{-1}\mathbf{f}\right)$$

data functionals solver forcing data

Gradient of misfit:

$$\frac{\partial \mathcal{J}}{\partial \theta} = -2\boldsymbol{P}^{\dagger}\, \boldsymbol{S}_{TL}^{\dagger}\Big|_{\theta}^{-1} \boldsymbol{L}^{\dagger} \Sigma_{\mathrm{d}\theta}^{-1} \left(\mathbf{d} - \boldsymbol{L}\boldsymbol{S}^{-1}\mathbf{f}\right)$$

$$\boldsymbol{P}^{\dagger} = \left[\frac{\partial \nu}{\partial \theta}\right]^{T} \left[\mathrm{diag}\left[\beta(\sigma)\right]\boldsymbol{B} \quad \mathrm{diag}\left[\boldsymbol{B}\left(\frac{\mathrm{du}}{\mathrm{dt}} - \beta(\sigma)\right)\right] \right]$$

Need the same operators (in particular, the adjoint solver) to compute gradients of the data misfit (e.g., to fit parameters with conjugate gradients)

Summary: Applications of Data Assimilation

- Keep model trajectory "on track"

Sequential "filtering" ; forecast

- optimal interpolation of data (sparse in space/time) with physics based covariance/constraints
- bring model and data into a common framework to study processes, test hypotheses

Smoother; retrospective synthesis

Summary: Data Assimilation Implementation

- (Ensemble methods)
- Variational methods: gradient based optimization of penalty functional (c.f., geophysical inverse theory)

Need to implement

- → linearization of dynamics
- → adjoints
- → data functionals
- → optimization algorithms

Summary: Some CIG Issues (variational bias)

Plan ahead in coding the forward problem!

→ **Modular codes simplify adjoint development**

→ **Clear definition of input/output states is essential**

→ **Dependence on physical parameters should be explicit**

Support development of modules for

→**Observation functionals (*L*)**

→ **Physical parameters ($\nu(\theta)$)**

Support for modular optimization systems?