Computational methods in applied inverse problems

Uri Ascher

Department of Computer Science
University of British Columbia
October 2017
Four lectures

- Calibration and simulation of deformable objects
- **Data manipulation and completion**
- Estimating the trace of a large implicit matrix and applications
- Numerical analysis in visual computing: not too little, not too much
Outline

- Motivation
- Completing scarce (sparse) data
- Uncertainty in data locations
- Forced to cut corners
- Randomized algorithms for many data sets
- Conclusions
Here is the T-shirt

**When All Else Fails**

manipulate the data
The practice of manipulating given observed data for solving inverse problems is known to have its perils: loss of statistical relevance, danger of calibrating a model to handle our own generated errors, etc.

And yet it seems to be everywhere in practice!

1. “Completing scarce data” by some interpolation/extrapolation or other approximation
2. Preferring to see data given at regular mesh nodes, or otherwise having a hidden uncertainty in the location of data values
3. “Completing data” to obtain a more efficient algorithm
4. “Completing data” to obtain a “more solid theory”
5. Manipulating data because we don’t know how to solve the problem otherwise.

When is it OK to do this?!

Attempt to get more insight by considering case studies.
Data completion and manipulation

The practice of manipulating given observed data for solving inverse problems is known to have its perils: loss of statistical relevance, danger of calibrating a model to handle our own generated errors, etc.

And yet it seems to be everywhere in practice!

1. “Completing scarce data” by some interpolation/extrapolation or other approximation
2. Preferring to see data given at regular mesh nodes, or otherwise having a hidden uncertainty in the location of data values
3. “Completing data” to obtain a more efficient algorithm
4. “Completing data” to obtain a “more solid theory”
5. Manipulating data because we don’t know how to solve the problem otherwise.

When is it OK to do this?!

Attempt to get more insight by considering case studies.
Data completion and manipulation

- The practice of manipulating given observed data for solving inverse problems is known to have its perils: loss of statistical relevance, danger of calibrating a model to handle our own generated errors, etc.
- And yet it seems to be everywhere in practice!
  1. "Completing scarce data" by some interpolation/extrapolation or other approximation
  2. Preferring to see data given at regular mesh nodes, or otherwise having a hidden uncertainty in the location of data values
  3. "Completing data" to obtain a more efficient algorithm
  4. "Completing data" to obtain a "more solid theory"
  5. Manipulating data because we don’t know how to solve the problem otherwise.

- When is it OK to do this?!
- Attempt to get more insight by considering case studies.
Our case studies

1. Inverting Maxwell’s equations and DC resistivity in exploration geophysics
   [Haber, A. & Oldenburg, 2004]

2. Recovering local volatility surface in financial mathematics

3. Denoising of surface triangle mesh
   [Huang & A., 2008]

4. Calibrating and simulating soft bodies in computer graphics
   [Wang, Wu, Yin, A., Liu & Huang, 2015]

5. Obtaining union of observation locations for many data sets
   [Roosta-Khorasani, van den Doel & A., 2014]
**Inverse problem setting**

- Given **observed data** \( d \in \mathbb{R}^l \) and a **forward operator** \( f(m) \) which provides **predicted data** for each instance of distributed parameter function \( m \), find \( m \) (discretized and reshaped into \( m \)) such that the predicted and observed data agree to within noise:

\[
d = f(m) + \eta.
\]

- Consider a case where a PDE must be solved to evaluate the forward operator, i.e., \( f(m) = Pu = PG(m)q \), where \( G \) is a discrete Green’s function.

- Iterative algorithm on \( m \) to reduce objective. Assuming \( \eta \sim \mathcal{N}(0, \sigma^2 I) \), the maximum likelihood (ML) **data misfit function** is

\[
\phi(m) = \|f(m) - d\|^2_2
\]

The discrepancy principle yields the stopping criterion

\[
\phi(m) \leq \rho, \quad \text{where} \quad \rho = \sigma^2 l
\]
**Inverse problem setting**

- Given observed data \( d \in \mathbb{R}^l \) and a forward operator \( f(m) \) which provides predicted data for each instance of distributed parameter function \( m \), find \( m \) (discretized and reshaped into \( m \)) such that the predicted and observed data agree to within noise:

\[
d = f(m) + \eta.
\]

- Consider a case where a PDE must be solved to evaluate the forward operator, i.e., \( f(m) = Pu = PG(m)q \), where \( G \) is a discrete Green’s function.

- Iterative algorithm on \( m \) to reduce objective. Assuming \( \eta \sim \mathcal{N}(0, \sigma^2 I) \), the maximum likelihood (ML) data misfit function is

\[
\phi(m) = \|f(m) - d\|_2^2
\]

The discrepancy principle yields the stopping criterion

\[
\phi(m) \leq \rho, \quad \text{where} \quad \rho = \sigma^2/l
\]
Inverse problem setting

- Given observed data $d \in \mathbb{R}^l$ and a forward operator $f(m)$ which provides predicted data for each instance of distributed parameter function $m$, find $m$ (discretized and reshaped into $m$) such that the predicted and observed data agree to within noise:

$$d = f(m) + \eta.$$

- Consider a case where a PDE must be solved to evaluate the forward operator, i.e., $f(m) = Pu = PG(m)q$, where $G$ is a discrete Green’s function.

- Iterative algorithm on $m$ to reduce objective. Assuming $\eta \sim \mathcal{N}(0, \sigma^2 I)$, the maximum likelihood (ML) data misfit function is

$$\phi(m) = \|f(m) - d\|_2^2$$

The discrepancy principle yields the stopping criterion

$$\phi(m) \leq \rho, \quad \text{where} \quad \rho = \sigma^2 l.$$
Outline

- Motivation
- Completing scarce (sparse) data
- Uncertainty in data locations
- Forced to cut corners
- Randomized algorithms for many data sets
- Conclusions
Example (CS1): scarce data in electromagnetic data inversion

$G$ is Green’s function for Maxwell’s equations in time or frequency domain, $m$ is conductivity or resistivity. [Haber, A. & Oldenburg, 2004]
EM DATA INVERSION IN GEOPHYSICS

Use Tikhonov-type regularization: a prior penalizing lack of smoothness in surface function $m$ through gradient.

Top: misfit. Bottom: recovered $m$. 
**Dupire’s equation (CS2)**

[Dupire, 1994]: replace the Black-Scholes equation for option price by a parabolic PDE of the form

\[
\frac{\partial C}{\partial \tau} = \frac{1}{2} \sigma^2(\tau, K) K^2 \frac{\partial^2 C}{\partial K^2} - bK \frac{\partial C}{\partial K}, \quad \tau > 0, K \geq 0,
\]

s.t. initial and boundary conditions (for calls)

\[
C(\tau = 0, K) = (S_0 - K)^+, \\
\lim_{K \to \infty} C(\tau, K) = 0, \quad \lim_{K \to 0} C(\tau, K) = S_0.
\]

Here \(\tau\) is time to maturity, \(K\) is strike price, \(C = C(\tau, K)\) is value of the European call option with expiration date \(T = \tau\), and \(\sigma(\tau, K)\) is volatility. Can write all this in operator form as

\[
\tilde{L}(\sigma) C = \tilde{q}(S_0),
\]

with \(\tilde{L}\) a linear differential operator for a given \(\sigma\).

Assume first that the stock price \(S_0\) is a given parameter.

**Calibrating the model:** solve inverse problem for \(\sigma(\tau, K)\) given \(C\)-data.
To simplify and make problem dimensionless, change \( K \) to \( y = \log(K/S_0) \) (so \( -\infty < y < \infty \)), then \( u(\tau, y) = C(\tau, S_0 \exp(y))/S_0 \) and \( m(\tau, y) = \frac{1}{2} \sigma(\tau, K(y))^2 \). Obtain

\[
-\frac{\partial u}{\partial \tau} + m \left( \frac{\partial^2 u}{\partial y^2} - \frac{\partial u}{\partial y} \right) + b \frac{\partial u}{\partial y} = 0, \quad \tau > 0, \ y \in \mathbb{R},
\]

s.t. side conditions

\[
u(\tau = 0, y) = (1 - \exp(y))^+, \quad \lim_{y \to \infty} u(\tau, y) = 0, \quad \lim_{y \to -\infty} u(\tau, y) = 1.
\]

Can write this as \( L(m)u = q \).
Discretize over a mesh with step sizes \( \Delta \tau \) and \( \Delta y \); denote the corresponding discretization as

\[
L_h(m)u_h = q_h.
\]
**Example: Locations of (real) data**

Locations of $u$-data values for the PBR data set.
EXAMPLE: LOCATIONS OF (REAL) DATA

Locations of $u$-data values for the SPX data set.
Interpolating/extrapolating the data

- Several researchers have applied interpolation/extrapolation to this type of data, followed by assimilation of the resulting data set with the discretized Dupire PDE problem.
- Use [Kahale, 2005] for this purpose. This algorithm applies data completion with a “financial prior”, insisting that the resulting data surface reproduce the “smile” effect.
- An obvious objection, however, is that the resulting data surface does not satisfy the discretized differential problem for any \( m(\tau, y) \), and vice versa. The assimilation of these two pieces of information may be more difficult.
- Compare this to not modifying the given data, using for both cases a Tikhonov-type regularization as well as EnKF.
Several researchers have applied interpolation/extrapolation to this type of data, followed by assimilation of the resulting data set with the discretized Dupire PDE problem.

Use [Kahale, 2005] for this purpose. This algorithm applies data completion with a “financial prior”, insisting that the resulting data surface reproduce the “smile” effect.

An obvious objection, however, is that the resulting data surface does not satisfy the discretized differential problem for any \( m(\tau, y) \), and vice versa. The assimilation of these two pieces of information may be more difficult.

Compare this to not modifying the given data, using for both cases a Tikhonov-type regularization as well as EnKF.
**Regularized Inverse Problem**

- Maximum likelihood for simplest case of white noise:

\[
\phi(m_h, u_h) = \|Pu_h - d\|^2 = \|PL_h(m_h)^{-1}q_h - d\|^2
\]

where matrix \(P\) projects to data locations: \(P\) has many more columns than rows for original data, whereas \(P = I\) for completed data.

- Regularize the problem: minimize the maximum a posteriori (MAP) merit function

\[
\phi_R(m_h, u_h) = \phi(m_h, u_h) + R(m_h).
\]

- Our Tikhonov-like regularization operator is \((a_0\) a known constant)\n
\[
R(m_h) = \alpha_1 \sum_i \sum_j (m_{i,j} - a_0)^2
\]

\[
+ \frac{\alpha_2}{\Delta \tau^2} \sum_j \sum_{i=1}^{M_{\tau}} (m_{i,j} - m_{i-1,j})^2 + \frac{\alpha_3}{\Delta y^2} \sum_j \sum_{i=1}^{M_y} (m_{i,j} - m_{i,j-1})^2
\]
Set $\alpha_1 > 0$ and compare working with the given sparse data vs using data completion by the Kahale algorithm.
Conclusion for case study CS2

- These results clearly show that the data completion approach has not delivered.

- Additional tests for Henry Hub and WTI prices, using bilinear interpolation for the data completion and different $\alpha$-weights in the Tikhonov-type priors, also clearly indicate that it is better to avoid the extensive data completion required here: the market implied smile, which has an important relationship with market risk, is better fitted upon using just the original data.

- Both EnKF algorithms we tried [Iglesias, Law & Stuart, 2013; Calvetti, Ernst & Somersalo 2014] were trivially (and significantly) improved by adding additional regularization using $a_0$ and first derivatives.

- After this improvement the EnKF algorithms were comparable to but not better than the Tikhonov-type regularization. Big plus: no ad hoc parameter search was required.
Conclusion for case study CS2

● These results clearly show that the data completion approach has not delivered.

● Additional tests for Henry Hub and WTI prices, using bilinear interpolation for the data completion and different $\alpha$-weights in the Tikhonov-type priors, also clearly indicate that it is better to avoid the extensive data completion required here: the market implied smile, which has an important relationship with market risk, is better fitted upon using just the original data.

● Both EnKF algorithms we tried [Iglesias, Law & Stuart, 2013; Calvetti, Ernst & Somersalo 2014] were trivially (and significantly) improved by adding additional regularization using $a_0$ and first derivatives.

● After this improvement the EnKF algorithms were comparable to but not better than the Tikhonov-type regularization. Big plus: no ad hoc parameter search was required.
Conclusion for case study CS2

- These results clearly show that the data completion approach has not delivered.
- Additional tests for Henry Hub and WTI prices, using bilinear interpolation for the data completion and different $\alpha$-weights in the Tikhonov-type priors, also clearly indicate that it is better to avoid the extensive data completion required here: the market implied smile, which has an important relationship with market risk, is better fitted upon using just the original data.
- Both EnKF algorithms we tried [Iglesias, Law & Stuart, 2013; Calvetti, Ernst & Somersalo 2014] were trivially (and significantly) improved by adding additional regularization using $a_0$ and first derivatives.
- After this improvement the EnKF algorithms were comparable to but not better than the Tikhonov-type regularization. Big plus: no ad hoc parameter search was required.
Motivation

Completing scarce (sparse) data

Uncertainty in data locations

Forced to cut corners

Randomized algorithms for many data sets

Conclusions
The notion that a (potentially noisy) data value $d_i$ is given at a known, deterministic location $x_i$ is often violated in practice. Here are some examples:

- **Surface triangle mesh denoising**  
  [Huang & A., 2008]

- **Minimal prospectivity mapping**  
  [Granek & Haber, 2014]  
  (not considered further)

- **Local volatility surface with uncertainty in price $S_0$**  
  [Albani, A., Yang & Zubelli, 2015]
Typical image denoising (CS3)

Left: noisy image: noisy data at precisely prescribed pixel locations.
Right: exact (ideally denoised?) image.
Surface triangle mesh denoising

Left: noisy triangle mesh: the data are nodal values \((x_i, y_i, z_i)\). No distinction between data value and location! Uncertainty in higher dimension.
Right: Our denoised triangle mesh.

[We had set out to generalize multiscale techniques for image denoising and ended up devising a completely different multiscale method for the surface mesh.]
Surface triangle mesh denoising

Left: noisy triangle mesh: the data are nodal values \((x_i, y_i, z_i)\). No distinction between data value and location! Uncertainty in higher dimension.

Right: Our denoised triangle mesh.

[We had set out to generalize multiscale techniques for image denoising and ended up devising a completely different multiscale method for the surface mesh.]
Scanned noisy model (25K verts)
Smoothed by MSAL
Uncertainty in $S_0$ (CS2)

- Recall Dupire’s equation

\[
\frac{\partial C}{\partial \tau} = \frac{1}{2} \sigma^2(\tau, K) K^2 \frac{\partial^2 C}{\partial K^2} - bK \frac{\partial C}{\partial K}, \quad \tau > 0, K \geq 0,
\]

\[
C(\tau = 0, K) = (S_0 - K)^+, \quad \lim_{K \to \infty} C(\tau, K) = 0, \quad \lim_{K \to 0} C(\tau, K) = S_0.
\]

But now, stock price $S_0$ has (well-quantified) uncertainty, as it is typically some average of daily prices.

- Add a term $\alpha_5 (S_0 - \hat{S}_0)^2$ to regularization prior $R = R(m_h, S_0)$, where $S_0$ is now random variable with measured mean (say) $\hat{S}_0$. 
**Uncertainty in \( S_0 \) cont.**

- Furthermore, recall that upon changing variables to log moneyness 
  \[ y = \log(K/S_0) \] (also \( u(\tau, y) = C(\tau, S_0 \exp(y))/S_0 \), 
  \[ m(\tau, y) = \frac{1}{2} \sigma(\tau, K(y))^2 \]), obtain the nicer PDE 

\[
- \frac{\partial u}{\partial \tau} + m \left( \frac{\partial^2 u}{\partial y^2} - \frac{\partial u}{\partial y} \right) + b \frac{\partial u}{\partial y} = 0, \quad \tau > 0, y \in \mathbb{R}.
\]

Now the uncertainty in \( S_0 \) has moved into the independent variable \( y \)!

- \( \Rightarrow \) In addition to adding a term \( \alpha_5 (S_0 - \hat{S}_0)^2 \) to \( R = R(m_h, S_0) \), update also

\[
\phi(m_h, S_0) = \alpha_0 \sum_{i=1}^{l} ((P(S_0)L_h(m_h)^{-1}q_h(S_0))_i - d_i)^2
\]

\[
+ \alpha_4 \sum_{j=1}^{M_y} \left( (1 - \exp(y_j(S_0)))^+ - (1 - \exp(y_j(\hat{S}_0)))^+ \right)^2.
\]

(In practice set \( \alpha_5 = 0 \).)
Uncertainty in $S_0$ cont.

- **Splitting method**: Alternately freeze $S_0$ and $m_h$ while solving for the other one.
- This converges fast because of the weak coupling, even when using the variable $y$.
- In preliminary experiments we see roughly a 1–2% change in adjusted price.
Outline

- Motivation
- Completing scarce (sparse) data
- Uncertainty in data locations
- Forced to cut corners?
- Randomized algorithms for many data sets
- Conclusions
In some cases, it has been claimed that data completion/manipulation allows obtaining better results. Here are some examples:

- Matrix and tensor completion for seismic applications

- Motion calibration and simulation of a soft body
  [Wang, Wu, Yin, A., Liu & Huang, 2015]
**Example: Full Waveform Inversion**

Herrmann: use data completion of velocity field in order to solve this problem.
Physically-based deformation modelling in Computer Graphics

Want to simulate and animate motion of a soft body, such as a plant under wind or water pressure, cloth, steak, face, etc.

Can model by elastodynamics and porous media, but need to calibrate the model.

Do that calibration by fitting example data obtained by sensor hardware: motion capturing and tracking.

[Wang, Wu, Yin, A., Liu & Huang; siggraph '15]
**Capturing data**

Left: three Kinect sensors are placed around the object; Right: the deformation point cloud sequence is captured at 30 Hz.
Capturing data cont.

Left: a (high resolution) surface mesh $\mathcal{S}$ with 15,368 vertices is used as a template to track captured point clouds; Right: its (low resolution) corresponding volumetric mesh $\mathcal{T}$ with 9,594 nodes is used for spatial co-rotated linear FEM simulations.
**Elastic Deformation**

- Denote reference shape by \( \mathbf{X} \) and dynamic, deformed positions at a time instant \( t \) by \( \mathbf{x} = \mathbf{x}(t) \).
- Element-wise stress-strain relationship using Hooke's law and Cauchy's linear strain tensor is
  \[
  \sigma = \mathbf{E} \varepsilon = \mathbf{E} \mathbf{B}_e (\mathbf{x}_e - \mathbf{X}_e),
  \]
  where the \( 6 \times 12 \) matrix \( \mathbf{B}_e = \mathbf{B}_e(\mathbf{X}_e) \) depends on \( \mathbf{X}_e \) nonlinearly.
- For isotropic materials the \( 6 \times 6 \) matrix \( \mathbf{E} \) only depends on Young's modulus \( E \) and Poisson's ratio \( \nu \).
- Denoting the per-element rotation matrix obtained from polar decomposition by \( \mathbf{R}_e = \mathbf{R}_e(\mathbf{x}_e(t), \mathbf{X}_e) \), the element-wise elastic forces using the co-rotated linear approximation are
  \[
  \mathbf{f}_e(E, \nu, \mathbf{X}_e, \mathbf{x}_e(t)) = \mathbf{R}_e \mathbf{K}_e (\mathbf{R}_e^T \mathbf{x}_e(t) - \mathbf{X}_e),
  \]
  \[
  \mathbf{K}_e = V_e \mathbf{B}_e^T \mathbf{E} \mathbf{B}_e,
  \]
  \( \mathbf{K}_e \) is \( 12 \times 12 \) element stiffness matrix and \( V_e \) is element volume.
Equations of Motion

- Assemble force contributions from all FEM elements
- Summon Newton’s 2nd law: at time $t$

\[
M\ddot{x} + D\dot{x} + \hat{K}x = (RK)x + f_{\text{ext}}, \quad \hat{K} = RKR^T.
\]

- Stiffness matrix $\hat{K}$ is sparse and is assembled from element contributions. Mass matrix $M$ is lumped.
- Use Rayleigh damping: $D = \alpha M + \beta \hat{K}$.
- Model calibration parameters in simplest case are

\[
p = (E, \nu, \alpha, \beta), \quad \text{so} \quad m = (p, X).
\]

Often have more than one control point, for each of which there is a Young modulus.
**Motion tracking and inverse problem**

- **Motion tracking**: physically-based probabilistic tracking: for given $p$ and $X$, find captured trajectory $x = \hat{x}_t$.
- This tracking problem involves an inference (EM) algorithm.
- **Inverse problem**: deformation parameter estimation:

\[
\min_{p, X} \sum_t \left\| x_t - \hat{x}_t \right\|^2.
\]
Employ a splitting method between $\mathbf{p}$ and $\mathbf{X}$. Fortunately this works very well because of weak coupling between these unknown groups.

Use Nelder-Mead for $\mathbf{p}$ as there are many local minima. However, this is good only for a few unknowns.

Many more nontrivial details are described in the paper.

Nice looking results are obtained! see the videos and a separate talk.

However, the calibration (parameter estimation) part of the process uses manufactured data.

Attempts to estimate the parameters directly from the point cloud data have not worked out well.
Data manipulation to the rescue

- Employ a splitting method between $p$ and $X$. Fortunately this works very well because of weak coupling between these unknown groups.
- Use Nelder-Mead for $p$ as there are many local minima. However, this is good only for a few unknowns.
- Many more nontrivial details are described in the paper.
- Nice looking results are obtained! see the videos and a separate talk.
- However, the calibration (parameter estimation) part of the process uses manufactured data.
- Attempts to estimate the parameters directly from the point cloud data have not worked out well.
Outline

- Motivation
- Completing scarce (sparse) data
- Uncertainty in data locations
- Forced to cut corners
- Randomized algorithms for many data sets
- Conclusions
### Inverse Problem with \( s \) Data Sets (CS5)

After discretization and for our problems of interest:

\[
\begin{align*}
    d_i &= f_i(m) + \eta_i, \quad i = 1, 2, \ldots, s \\
    f_i(m) &= Pu_i = PG(m)q_i
\end{align*}
\]

Calculating "\( G(m)q_i \)" for each \( i \) is costly!

- \( d_i \in \mathbb{R}^l \) is the measurement obtained in the \( i^{th} \) experiment
- \( f_i \) is the known forward operator for the \( i^{th} \) experiment
- \( m \in \mathbb{R}^{lm} \) is the sought-after model
- \( \eta_i \) is the noise incurred in the \( i^{th} \) experiment
- \( s \) is the total number of experiments
- \( u_i \in \mathbb{R}^{lu} \) is the \( i^{th} \) field
- \( q_i \in \mathbb{R}^{lu} \) is the \( i^{th} \) source
- \( G^{-1} \) is a square matrix discretizing the PDE with the BC
- \( P = P_i \) is the projection matrix for the \( i^{th} \) experiment
**Application: DC resistivity**

- PDE with multiple sources
  \[
  \nabla \cdot (\mu(x)\nabla u_i) = q_i, \quad i = 1, \ldots, s,
  \]
  \[
  \frac{\partial u_i}{\partial \nu} |_{\partial \Omega} = 0.
  \]

- Conductivity $\mu(x)$ is expressed as a point-wise function of $m(x)$ (e.g., use tanh to incorporate known bounds on $\mu$).
- The operator $G(m)$ is the inverse of the above PDE discretized on a staggered grid.
- Use different selections of sources $q_i$, yielding corresponding fields $u_i$.
- Data is measured only on part of the domain’s boundary.
- Use any prior we may have for this very difficult problem!
**DC resistivity experiment setup**

- Domain $\Omega$ is the unit square. Sources are of the form
  \[ q_i(x) = \delta(x - x_1^i) - \delta(x - x_2^i) \]
  with $x_1$ positive unit point source on west boundary, $x_2$ negative unit
  point source on east boundary. Vary $p$ boundary wall locations to get
  $s = p^2$ data sets.
- Receivers are all grid points on north and south walls. No sources or
  receivers at corners.
- Uniform $64 \times 64$ mesh
- For bounds set $\mu_{\text{max}} = 1.2 \max \mu(x)$, $\mu_{\text{min}} = 1.2^{-1} \min \mu(x)$
- PCG inner iteration limit $r = 20$; cgto1 = $1.e-3$. 
**Example:** \( \mu_1 = .1, \mu_{ll} = 1, \mu_{lll} = .01, \text{ NOISE 2\%} \)

- Thus, we want \( s \) larger for better reconstruction quality.
- But the cost of solving the problem grows very fast! (at least linearly with \( s \)). Need to find more efficient approximations for evaluating misfit function \( \phi(m) \).
**Example:** $\mu_1 = .1$, $\mu_{11} = 1$, $\mu_{111} = .01$, **noise** 2%

- Thus, we want $s$ larger for better reconstruction quality.
- But the cost of solving the problem grows very fast! (at least linearly with $s$). Need to find more efficient approximations for evaluating misfit function $\phi(m)$. 
Monte Carlo to approximate the misfit trace

- Let $B(m) = F(m) - D \in \mathbb{R}^{l \times s}$. In $k$th iteration, $m = m_k$.
- Then $A = B^T B$ is implicit symmetric positive semi-definite (SPSD); effectively, can only carry out matrix-vector products $A \ast v$ with this $s \times s$ matrix.

$$\phi(m) = \|B(m)\|_F^2 = tr(B^T B) = \mathbb{E}(w^T A w).$$

- Approximating expectation $\Leftrightarrow$ Approximating the trace $\phi(m) = tr(A)$
  - Monte-Carlo approximation

$$tr(A) \approx \frac{1}{s_k} \sum_{j=1}^{s_k} w_j^T A w_j = \frac{1}{s_k} \sum_{j=1}^{s_k} \|B w_j\|_2^2.$$

- Note we can obtain exact trace using $s_k = s$ samples with $w_j$ a scaled $j$th column of identity; but we want $s_k \ll s$. 
**Using stabilized Gauss-Newton with total variation (TV) added**

<table>
<thead>
<tr>
<th>Method</th>
<th>Vanilla (3,969)</th>
<th>Gaussian (3,969)</th>
<th>Vanilla (49)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work</td>
<td>476,280</td>
<td>4,618</td>
<td>5,978</td>
</tr>
</tbody>
</table>

(g) True model  
(h) Gaussian, s=3,969  
(i) Vanilla, s=49

**BUT** what if $P = P_i$ varies with $i$, i.e., data for different experiments is not given at same locations?  
Can no longer write $\sum_{i=1}^{s} w_i PG(m)q_i = \sum_{i=1}^{s} PG(m)(w_iq_i)$, and the magic of the randomized algorithm is gone.
Using stabilized Gauss-Newton with total variation (TV) added

<table>
<thead>
<tr>
<th>Method</th>
<th>Vanilla (3,969)</th>
<th>Gaussian (3,969)</th>
<th>Vanilla (49)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work</td>
<td>476,280</td>
<td>4,618</td>
<td>5,978</td>
</tr>
</tbody>
</table>

(j) True model  
(k) Gaussian, s=3,969  
(l) Vanilla, s=49

BUT what if \( P = P_i \) varies with \( i \), i.e., data for different experiments is not given at same locations?

Can no longer write \( \sum_{i=1}^{s} w_i P G(m) q_i = \sum_{i=1}^{s} P G(m)(w_i q_i) \), and the magic of the randomized algorithm is gone.
**Data approximation methods**

Let receivers of \( i \)th data set be in \( \Gamma_i \subset \partial \Omega, \ i = 1, 2, \ldots, s \). Want to extend data to the union \( \Gamma = \bigcup \Gamma_i \subseteq \partial \Omega \).

1. DCT, wavelets, curvelets for each \( i \). Advantage: leverage the recent advances in compressive sensing and sparse \( \ell_1 \) methods.
2. Piecewise linear interpolation for each \( v_i \). Advantage: very simple.
3. L2G: data completion function \( v_i \in H^1(\Gamma) \) solves discretization of

\[
\min_v \| v - d_i \|_{L_2(\Gamma_i)}^2 + \beta \| \nabla S v \|_{L_2(\Gamma)}^2.
\]

4. Data completion function \( v_i \in H^2(\Gamma) \) solves discretization of

\[
\min_v \| v - d_i \|_{L_2(\Gamma_i)}^2 + \beta \| \Delta S v \|_{L_2(\Gamma)}^2.
\]

Which method to use?
Choosing data approximation method

- Use Mathematics, not Politics, to select method: this approach can work in practice.
- Concentrate on EIT / DC resistivity with piecewise smooth conductivity $\mu(x)$.

1. If $\mu$’s discontinuities are all away from boundary, then $u \in H^2(\Gamma)$. So, use regularization with Laplacian (Option 4).
2. If $\mu$ has discontinuities which extend to boundary, then $u \in H^1(\Gamma)$. So, use L2G (Option 3).

See [Roosta, Doel & A., 2014] for theorems justifying the above.
Choosing data approximation method

Experiments with 50\% data completion and 5\% noise:

Left: Laplacian for $u \in H^2(\Gamma)$. Right: L2G for $u \in H^1(\Gamma)$
$\mu_I = 0.1$, $\mu_{II} = 1$, NOISE 3%, $s = 961$, COMP. 20%

Compare using Gaussian distribution with data completion vs random subset which does not require data completion.

<table>
<thead>
<tr>
<th>Method</th>
<th>Random Subset</th>
<th>Data Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work</td>
<td>3,367</td>
<td>1,597</td>
</tr>
</tbody>
</table>

Left: true. Center: RS. Right: DC
\( \mu_1 = .1, \mu_{II} = 1, \text{NOISE} \ 3\%, \ s = 961, \text{COMP.} \ 20\% \)

Compare using Gaussian distribution with data completion vs random subset which does not require data completion.

<table>
<thead>
<tr>
<th>Method</th>
<th>Random Subset</th>
<th>Data Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work</td>
<td>6,302</td>
<td>2,769</td>
</tr>
</tbody>
</table>

Left: true. Center: RS. Right: DC
\[ \mu_1 = 1, \mu_2 = .1, \text{NOISE 5\%, } s = 961, \text{ COMP. 50\%} \]

Compare using Gaussian distribution with data completion vs random subset which does not require data completion.

<table>
<thead>
<tr>
<th>Method</th>
<th>Random Subset</th>
<th>Data Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work</td>
<td>5,139</td>
<td>2,320</td>
</tr>
</tbody>
</table>

Left: true. Center: RS. Right: DC
\( \mu_1 = 1, \mu_1 = .1,\) NOISE 5%, \( s = 961,\) COMP. 50%

Compare using Gaussian distribution with data completion vs random subset which does not require data completion.

<table>
<thead>
<tr>
<th>Method</th>
<th>Random Subset</th>
<th>Data Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work</td>
<td>5,025</td>
<td>1,818</td>
</tr>
</tbody>
</table>

Left: true. Center: RS. Right: DC
In the accompanying paper there are also 3D results, similar to the ones for the case with the same receivers.

Both variants with and without level set method were tried.

The Data Completion method was always faster than Random Subset by a factor of at least 2 and up to 4.

Data completion of up to $\approx 50\%$ works fine. But reconstructions deteriorate upon completing scarcer data!
OUTLINE

- Motivation
-Completing scarce (sparse) data
- Uncertainty in data locations
- Forced to cut corners?
- Randomized algorithms for many data sets
- Conclusions
Conclusions

- Data completion and other statistically unholy manipulations such as ignoring location uncertainty are **not an ideal undertaking** from a theoretical point of view.

- But in practical situations it is often **quietly done** by mathematicians, computer scientists and engineers alike.

- We have seen instances where (more massive) such practices **should be avoided**.

- We have seen instances where such practices **can be tolerated**, typically when other uncertainties dominate.

- We have seen instances where such practices seem **essential for obtaining plausible results**, and where better algorithms are further sought.

- The larger the proportion of missing data, the harder it is to produce an adequate completed set.
Bayes made me do it