Toolkits and Techniques for Debugging Inverse Problems

A Boyle[1], WRB Lionheart[2], A Adler[3]

[1] *University of Ottawa, Ottawa, Canada* [2] *University of Manchester, Manchester, UK* [3] *Carleton University, Ottawa, Canada*

Problem

Inverse problems lead to inherently low resolution images which can be difficult to verify. Regularization can stabilize solutions, but lead to bias. Typically, the algorithms successfully produce images, but the reconstructed images have artefacts which may lead to wrong interpretations. Our work is motivated by this "debugging" process, leading to two questions: How should one go about determining whether a result is satisfactory? And, if the result is wrong, what caused the failure?

In this work, we report our processes, and the techniques used to find issues in the specific context of impedance imaging and, more generally, for inverse problems. We focus on the algorithmic aspects: the challenges in validating inverse problem codes as well as their inputs and outputs [1].

Tools

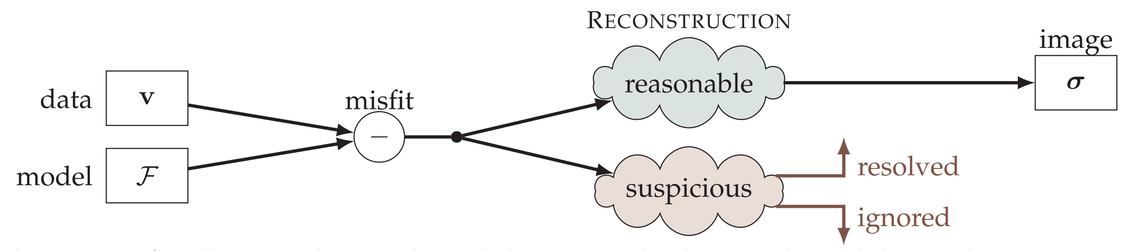

Figure 1 Explanations for data quality and model mismatch; data and model may have a mismatch which leads to a reasonable (green) or suspicious (red) reconstruction; a suspicious mismatch may be *resolved* and incorporated into the reconstruction or *ignored* by reducing its affect on the reconstruction

We wish to avoid conflating the reconstruction algorithm with the quality of the data. We initially rely solely on the forward model's correct implementation to examine the data (Figure 1). Data that agrees with the model to a certain degree will result in a "reasonable" reconstruction while a significant data-model mismatch may cause a "suspicious" artifact. A suspicious artifact may either be ignored, by discarding or de-weighting measurements, or the cause of the problem can be resolved by adjusting the model to make the data useful in the reconstruction.

Debugging

By developing these debug tools, we aim to improve the rigour and quality of image reconstruction in a comprehensive framework by enabling quick debug cycles that can feedback to improved data acquisition strategies and better interpretive outcomes.

This work is naturally related to theories of software and algorithm debugging [2], defect and root cause analysis [3] and business process analysis [4, 5]. Methods for debugging software or hardware are very heuristic by nature; they tend to be problem dependent.

We propose a set of tools to enable a "debugging" workflow for evaluating an inverse problem result. A systematic process such as this can help the dissection of an algorithm to locate a specific cause within the code base or algorithm inputs. Our workflow provides debugging tools at three stages for evaluating: 1) data and model quality, 2) algorithm behaviour, and 3) regional image fit.

Given the challenges inherent in demonstrating a correct or valid reconstruction, we propose that, for inverse problems in general, trust in the overall design may be constructed by testing units of functionality and then building upon that foundation. Individual components may (#1) be validated to perform as expected in isolation, or (#2) compared against some gold standard. Next, (#3) the components can be reassembled and simulated data followed through the algorithmic machinery; the outcome can be compared to a synthetic model which was used to construct the input data. Finally, (#4) the real data can be compared to simulated data.

Hierarchy

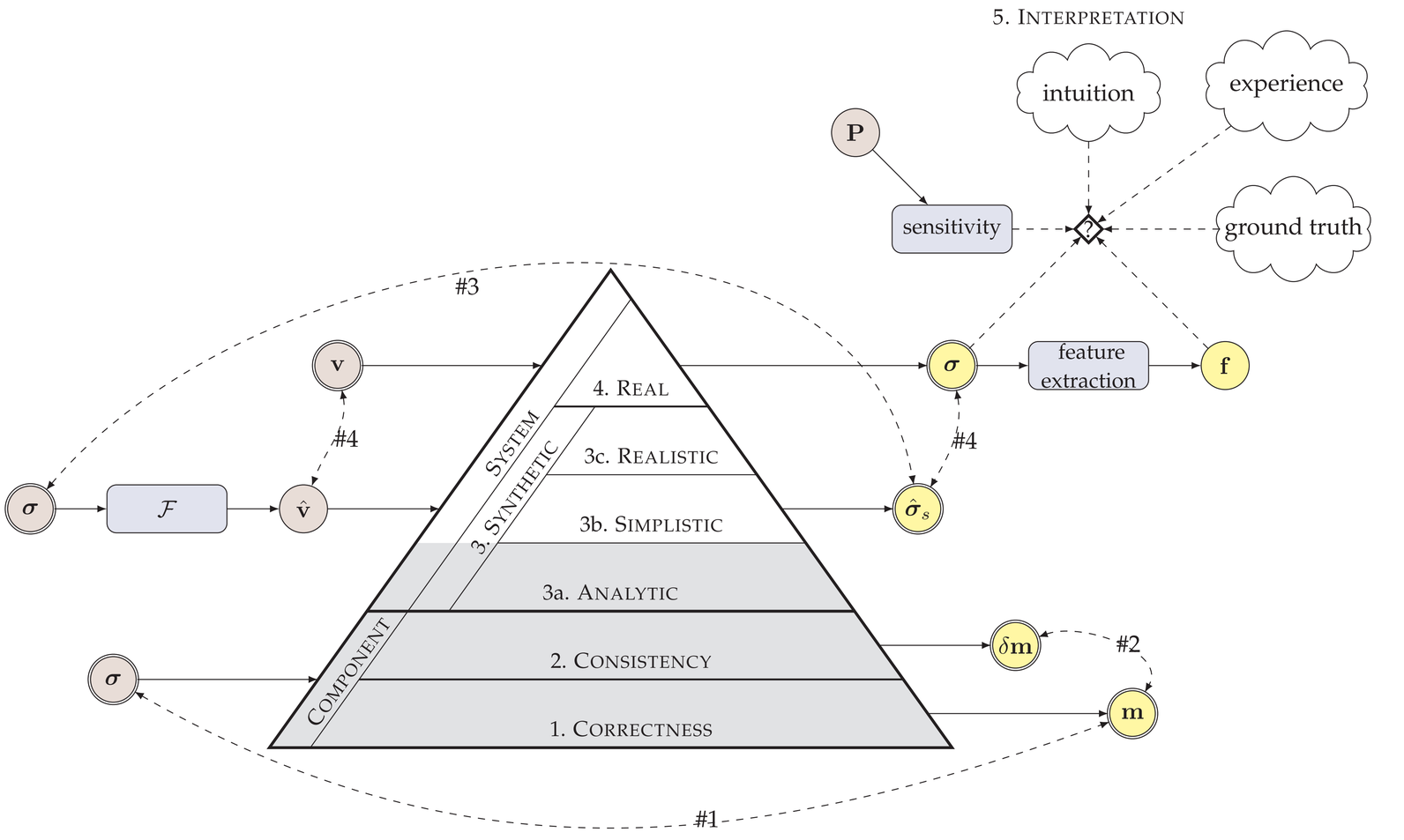

Figure 2 Proposed hierarchy of validation; proposed in this thesis, internal checks of components and system □ form a trusted platform upon which the input data may be compared to expectations □, an interpretation (#5) decides whether a result is reasonable, (#4) input ○ and output ○ may be compared, (#3) synthetic data may be used to confirm correctness of the outcome for (a) analytic, (b) simplistic and (c) realistic data, analytic solutions provide a mechanism for independently confirming forward model $\mathcal{F}$ correctness, (#2) consistency between related groups of components may be validated and (#1) from each set of components checked for consistency at least one needs to be confirmed to provide correct results, points of comparison are identified with dashed lines ← →

In this manner, a forward model using the finite element method may be validated against **resistor models** and **analytic** solutions [1]. By selecting appropriate models, the Jacobian may be validated by comparison against a **finite difference** perturbation. Data may be **plotted** to quickly identify outliers or **range conditions** may be tested based on noise estimates [6, 7].

A model can be manually refined to bring it into closer agreement with the real measurements. Model refinements may adapt the geometry, conductivity or other parameters to more closely match external data or hypotheses. If the two data sets appear reasonably aligned, the flow of the algorithm can be monitored between the two data sets to determine where processing diverges. The reconstructed image of the real data may be contrasted with other images, estimates of sensitivity, external information or extracted features (#5) to develop a meaningful interpretation of the image in context. These ideas are illustrated in Figure 2, inspired by Maslow's Hierarchy of Needs [8]. Ultimately, the desired outcome is to either (a) explain the unusual reconstruction as valid with a specific cause, (b) determine a mismatch in data quality, or (c) identify some flaw in the algorithm or its implementation. One would expect that, after some period of time with a stable code base, outcomes (a) and (b) would be predominant.

References

[1] A. Boyle. Ph.D. thesis, Carleton University, Ottawa, Canada (2016).

[2] S. Gill. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 206(1087) (1951) 538.

[3] P. Wilson, L. Dell, G. Anderson. *Root cause analysis: A tool for total quality management*. ASQ Quality Press (1993).

[4] T. Ohno. *Toyota Production System: beyond large-scale production*. Productivity Press (1988).

[5] M. Harry, R. Schroeder. *Six Sigma: The Breakthrough Management Strategy Revolutionizing the World's Top Corporations*. Random House (2000).

[6] F. John. *Duke Mathematical Journal*, 4(2) (1938) 300.

[7] W. R. B. Lionheart, K. Paridis, A. Adler. In *EIT2012*. Tianjin, China (2012).

[8] A. Maslow. *Psychological Review*, 50(4) (1943) 370.