

LA-UR- 11-05764

Approved for public release;  
distribution is unlimited.

Title: Time Series Analysis

Author(s): James L. Carroll  
Christ D. Tomkins

Intended for: 2011 BIE Users Group Working Group Presentation



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

## Time Series Analysis

2011 BIE Users Group

James L. Carroll

Chris D. Tomkins



Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

UNCLASSIFIED



### Abstract

- We use the S-BIE to perform a systematic study of time series analysis, using multiple noise regimes, and temporal degrees of freedom. Analyzing quality of the result obtained (and comparing that quality to that obtained with a static analysis). We are also interested in the optimization difficulty under the static and time series situations.



Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

UNCLASSIFIED

Slide 2



## The Bayesian Inference Engine is:

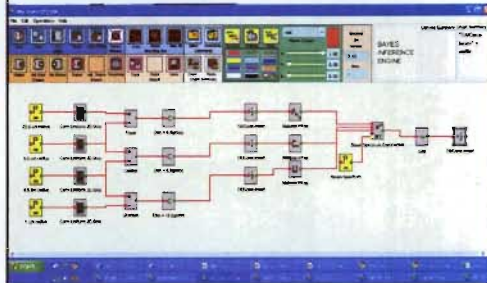
- The Lab's primary toolkit for DARHT Radiography Analysis:

## The BIE Consists of:

- A set of tools for radiographic forward modeling and parameter optimization
- A functional reactive graphic programming language for interfacing with those tools

# Comparison

## Graphical



- Intuitive
- Interactive

## Scripting

```

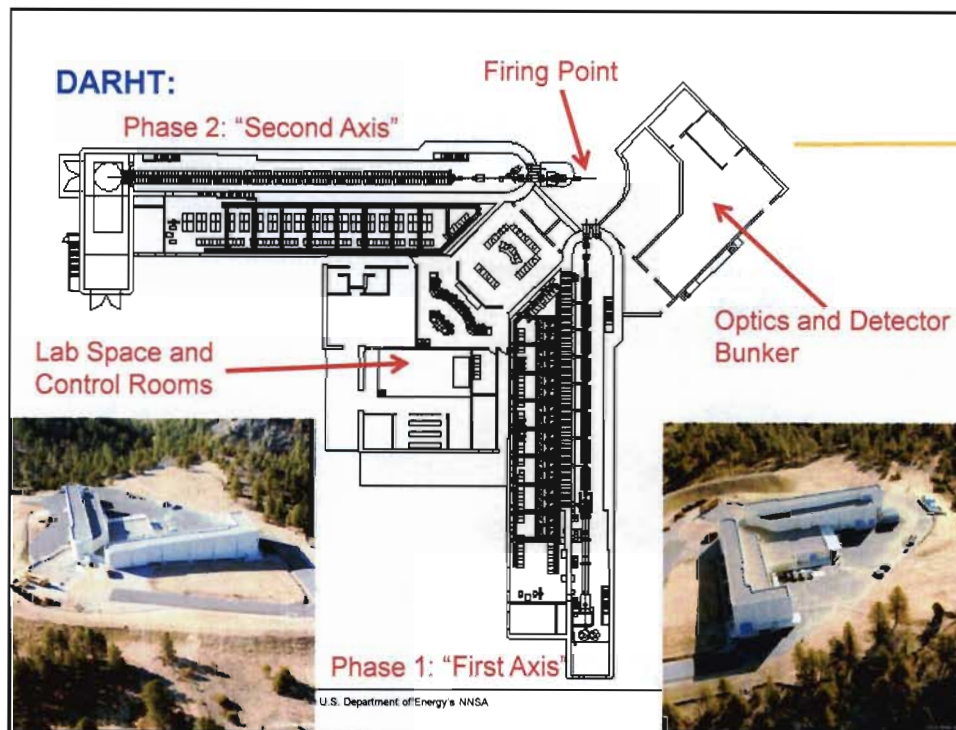
model = Model()
model.addBlock('Control', ControlType, VectorInputOutput, Input, Output)
model.addBlock('Process', ProcessType, VectorInputOutput, Input, Output)
model.addBlock('Output', OutputType, VectorInputOutput, Input, Output)
model.addBlock('Feedback', FeedbackType, VectorInputOutput, Input, Output)
model.addBlock('Sum', SumType, VectorInputOutput, Input, Output)
model.addBlock('Gain', GainType, VectorInputOutput, Input, Output)
model.addBlock('Delay', DelayType, VectorInputOutput, Input, Output)
model.addBlock('Integrator', IntegratorType, VectorInputOutput, Input, Output)
model.addBlock('Scope', ScopeType, VectorInputOutput, Input, Output)
model.addBlock('Plot', PlotType, VectorInputOutput, Input, Output)
model.addBlock('Table', TableType, VectorInputOutput, Input, Output)
model.addBlock('File', FileType, VectorInputOutput, Input, Output)
model.addBlock('Web', WebType, VectorInputOutput, Input, Output)
model.addBlock('Database', DatabaseType, VectorInputOutput, Input, Output)
model.addBlock('Network', NetworkType, VectorInputOutput, Input, Output)
model.addBlock('Cloud', CloudType, VectorInputOutput, Input, Output)
model.addBlock('Mobile', MobileType, VectorInputOutput, Input, Output)
model.addBlock('IoT', IoTType, VectorInputOutput, Input, Output)
model.addBlock('AI', AIType, VectorInputOutput, Input, Output)
model.addBlock('ML', MLType, VectorInputOutput, Input, Output)
model.addBlock('DL', DLType, VectorInputOutput, Input, Output)
model.addBlock('NN', NNType, VectorInputOutput, Input, Output)
model.addBlock('RF', RFType, VectorInputOutput, Input, Output)
model.addBlock('SVM', SVMType, VectorInputOutput, Input, Output)
model.addBlock('DT', DTType, VectorInputOutput, Input, Output)
model.addBlock('GB', GBType, VectorInputOutput, Input, Output)
model.addBlock('XGBoost', XGBoostType, VectorInputOutput, Input, Output)
model.addBlock('LightGBM', LightGBMType, VectorInputOutput, Input, Output)
model.addBlock('CatBoost', CatBoostType, VectorInputOutput, Input, Output)
model.addBlock('Pytorch', PytorchType, VectorInputOutput, Input, Output)
model.addBlock('TensorFlow', TensorFlowType, VectorInputOutput, Input, Output)
model.addBlock('Keras', KerasType, VectorInputOutput, Input, Output)
model.addBlock('Jax', JaxType, VectorInputOutput, Input, Output)
model.addBlock('Flax', FlaxType, VectorInputOutput, Input, Output)
model.addBlock('Gym', GymType, VectorInputOutput, Input, Output)
model.addBlock('OpenAI', OpenAIType, VectorInputOutput, Input, Output)
model.addBlock('Stable-Baselines', StableBaselinesType, VectorInputOutput, Input, Output)
model.addBlock('Ray', RayType, VectorInputOutput, Input, Output)
model.addBlock('Dask', DaskType, VectorInputOutput, Input, Output)
model.addBlock('Spark', SparkType, VectorInputOutput, Input, Output)
model.addBlock('Hadoop', HadoopType, VectorInputOutput, Input, Output)
model.addBlock('Kafka', KafkaType, VectorInputOutput, Input, Output)
model.addBlock('Redis', RedisType, VectorInputOutput, Input, Output)
model.addBlock('MongoDB', MongoDBType, VectorInputOutput, Input, Output)
model.addBlock('Cassandra', CassandraType, VectorInputOutput, Input, Output)
model.addBlock('HBase', HBaseType, VectorInputOutput, Input, Output)
model.addBlock('Pig', PigType, VectorInputOutput, Input, Output)
model.addBlock('MapReduce', MapReduceType, VectorInputOutput, Input, Output)
model.addBlock('Tez', TezType, VectorInputOutput, Input, Output)
model.addBlock('Hive', HiveType, VectorInputOutput, Input, Output)
model.addBlock('Impala', ImpalaType, VectorInputOutput, Input, Output)
model.addBlock('Presto', PrestoType, VectorInputOutput, Input, Output)
model.addBlock('Trino', TrinoType, VectorInputOutput, Input, Output)
model.addBlock('Doris', DorisType, VectorInputOutput, Input, Output)
model.addBlock('ClickHouse', ClickHouseType, VectorInputOutput, Input, Output)
model.addBlock('Flink', FlinkType, VectorInputOutput, Input, Output)
model.addBlock('Kubernetes', KubernetesType, VectorInputOutput, Input, Output)
model.addBlock('Docker', DockerType, VectorInputOutput, Input, Output)
model.addBlock('Ansible', AnsibleType, VectorInputOutput, Input, Output)
model.addBlock('Chef', ChefType, VectorInputOutput, Input, Output)
model.addBlock('Puppet', PuppetType, VectorInputOutput, Input, Output)
model.addBlock('SaltStack', SaltStackType, VectorInputOutput, Input, Output)
model.addBlock('Vagrant', VagrantType, VectorInputOutput, Input, Output)
model.addBlock('Terraform', TerraformType, VectorInputOutput, Input, Output)
model.addBlock('Packer', PackerType, VectorInputOutput, Input, Output)
model.addBlock('Consul', ConsulType, VectorInputOutput, Input, Output)
model.addBlock('Vault', VaultType, VectorInputOutput, Input, Output)
model.addBlock('Nomad', NomadType, VectorInputOutput, Input, Output)
model.addBlock('Kubernetes', KubernetesType, VectorInputOutput, Input, Output)
model.addBlock('Docker', DockerType, VectorInputOutput, Input, Output)
model.addBlock('Ansible', AnsibleType, VectorInputOutput, Input, Output)
model.addBlock('Chef', ChefType, VectorInputOutput, Input, Output)
model.addBlock('Puppet', PuppetType, VectorInputOutput, Input, Output)
model.addBlock('SaltStack', SaltStackType, VectorInputOutput, Input, Output)
model.addBlock('Vagrant', VagrantType, VectorInputOutput, Input, Output)
model.addBlock('Terraform', TerraformType, VectorInputOutput, Input, Output)
model.addBlock('Packer', PackerType, VectorInputOutput, Input, Output)
model.addBlock('Consul', ConsulType, VectorInputOutput, Input, Output)
model.addBlock('Vault', VaultType, VectorInputOutput, Input, Output)
model.addBlock('Nomad', NomadType, VectorInputOutput, Input, Output)

```

- Powerful
- Reproducible

## Preliminary results and uses

- We have created a prototype scripting language (the S-BIE), and implemented several of the BIE's forward modeling tools within the new language.
- Scripting enables systematic studies
- We have applied this new scripting language to analyze several time series problems



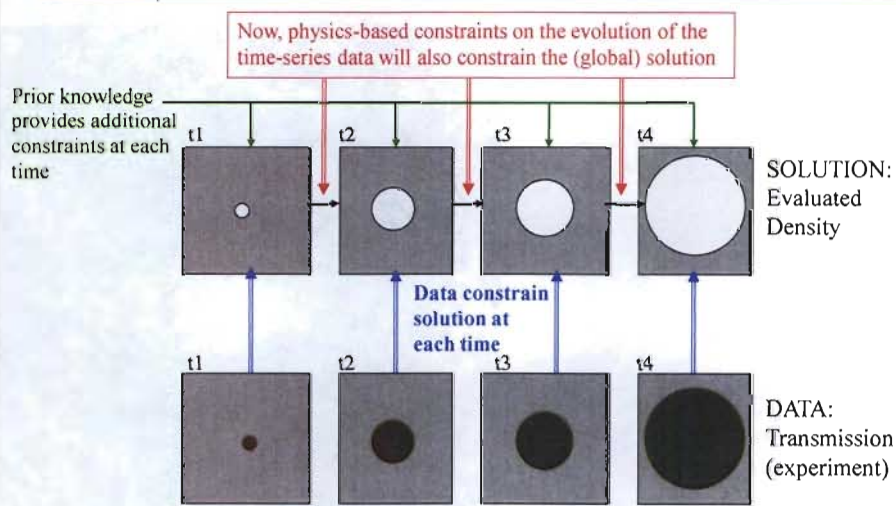
## DARHT Axis 2 Accelerator

- 2-ms, 2-kA, 18.4-MeV electron beam
- for 4-pulse radiography.
- Linear Induction Accelerator with wound Metglass cores and Pulse Forming Networks (PFNs) .
- The Injector uses a MARX bank with 88 type E PFN stages at 3.2 MV.
- Thermionic cathode.
- 4 micropulses - variable pulse width.
- Operations began in 2008.

## Time Series Analysis

- To date we have analyzed each element of DARHT axis 2 time series data independently
- But we should be able to use information in the time series to compensate for some of the noise, and get better results.

### The forward-modeling framework makes possible a global optimization procedure

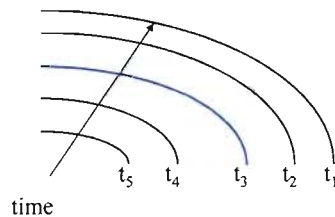


Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA



## These physics-based constraints will maximize information extracted from each dataset

Consider an evolving interface:



Concept: Can we learn something about the solution at time 3 (blue) from the data at surrounding times?

Approach: use physics to constrain solution at each time based upon time-series of data.

**WHEN WILL THIS APPROACH HAVE GREATEST VALUE?**

When certain conditions are met:

Data must be correlated in time!

→ 1) Must have the time between measurements ( $\Delta t$ ) on the order of a relevant time scale of the flow; and

Perfect data would be the only required constraint... (Noisier data means the global optimization adds more value).

→ 2) Must have non-perfect data (due to noise, background levels, etc).

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA



## Example

### ■ Graded Polygon:



**Los Alamos**  
NATIONAL LABORATORY  
EST. 1943

UNCL

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's

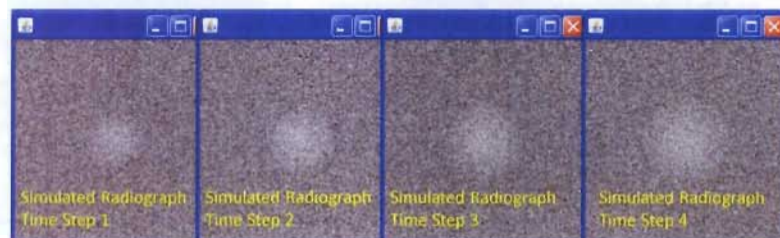


## S-BIE Simulated Expanding Object

- Object model:



- Simulated four time series radiographs:



## Explore Relationships Between:

- Degrees of temporal freedom
- Noise
- Optimization Difficulty



## Hypothesis 1:

- The complexity of the physics model (Degrees of temporal freedom)
  - Lower freedom provides more information from time series
  - Higher freedom provides less information from time series

## Hypothesis 2:

- Noise
  - Lower noise provides more static information
  - Higher noise provides less static information

## Hypothesis 3:

- Optimization Difficulty
  - Time series analysis can produce a more difficult optimization problem
  - The difficulty is related to the temporal degrees of freedom

## How to Model Physics

- There are many ways to model the time series
  - Simplest time series case:
    - Matching a polynomial of degree  $n$  (perfect physics)
  - More complex case:
    - Penalize divergence from some presumed smooth motion through time (imperfect physics)
- We will begin our analysis with the simplest case

## Radius Polynomials

- Degrees of Temporal Freedom:
  - 1:  $r = \text{knownConst} + p_1 t$
  - 2:  $r = p_0 + p_1 t$
  - 3:  $r = p_0 + p_1 t + p_2 t^2$
  - 4:  $r = p_0 + p_1 t + p_2 t^2 + p_3 t^3$
  - ...
  - n:  $r = p_0 + p_1 t + p_2 t^2 + p_3 t^3 + \dots + p_{n-1} t^{n-1}$

## Signal to Noise

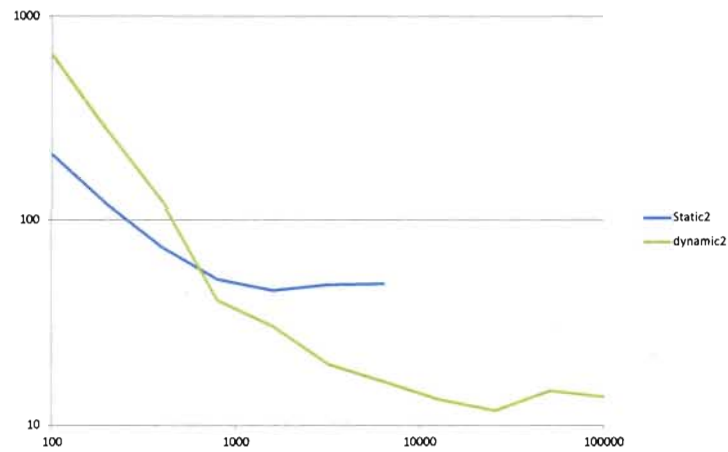
- Signal
  - 15 in center
  - 1 at edge
- Noise
  - Std deviations



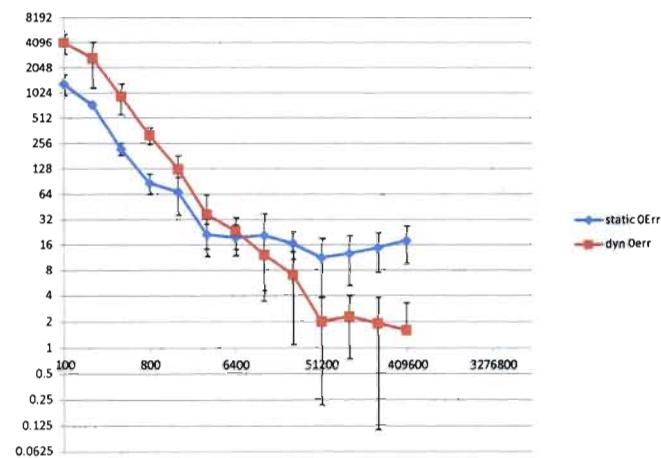
## Results

1 Degree of Temporal Freedom

## 2 sdev noise



## 0 noise



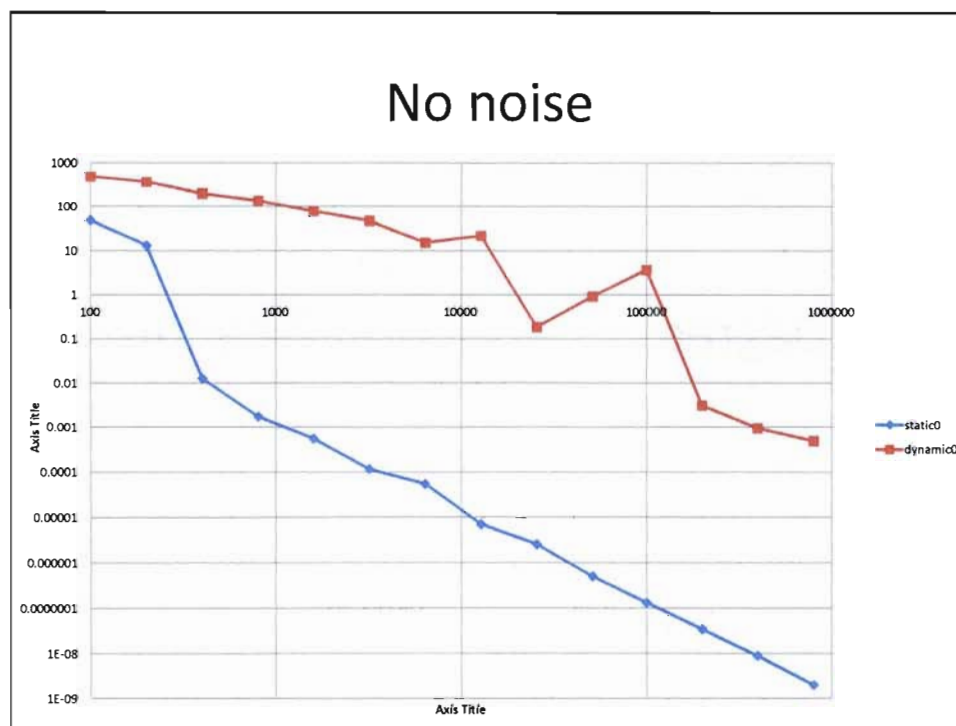
## 1 Degree of Temporal Freedom

- Results
  - Time Series Analysis is best for all noise ranges considered
  - Time Series Analysis is slightly more difficult for the optimizer

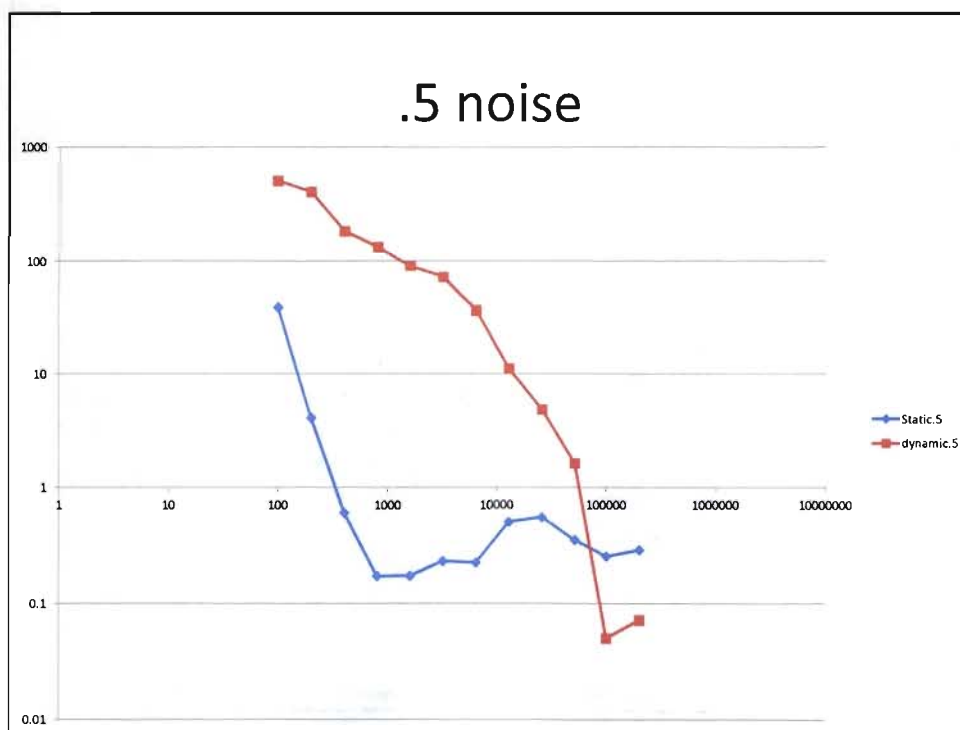
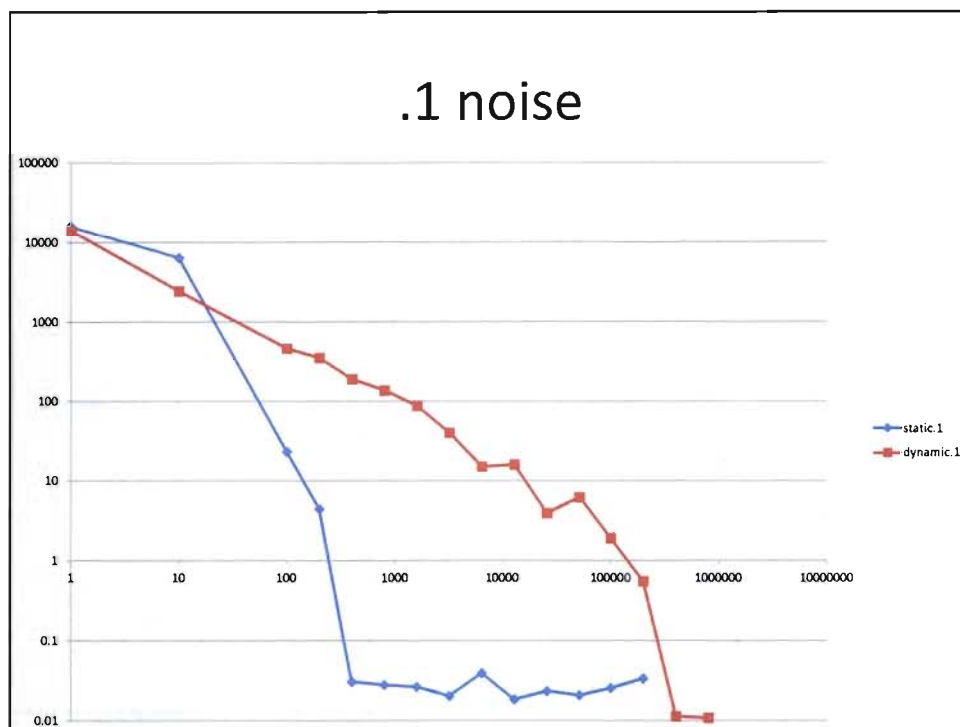
## 2 Degrees of Temporal Freedom

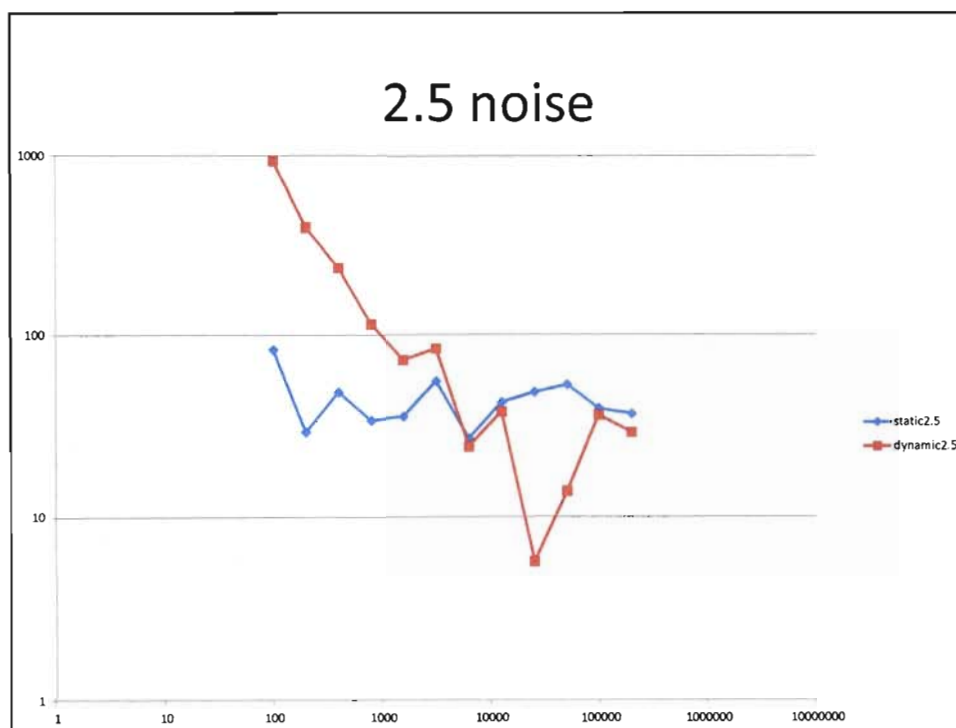
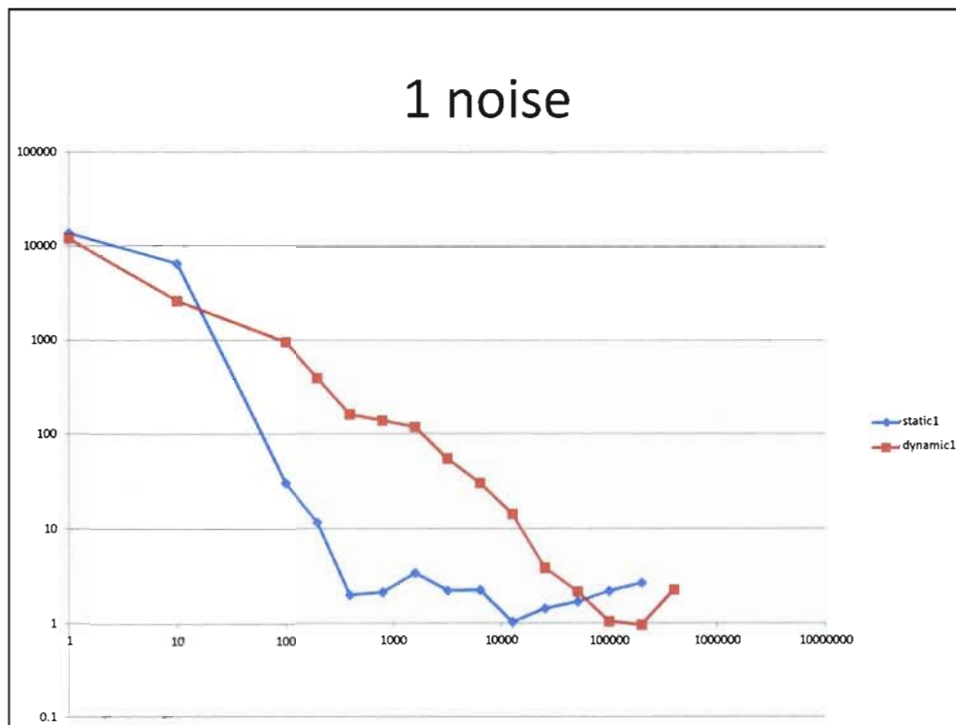
## Prediction

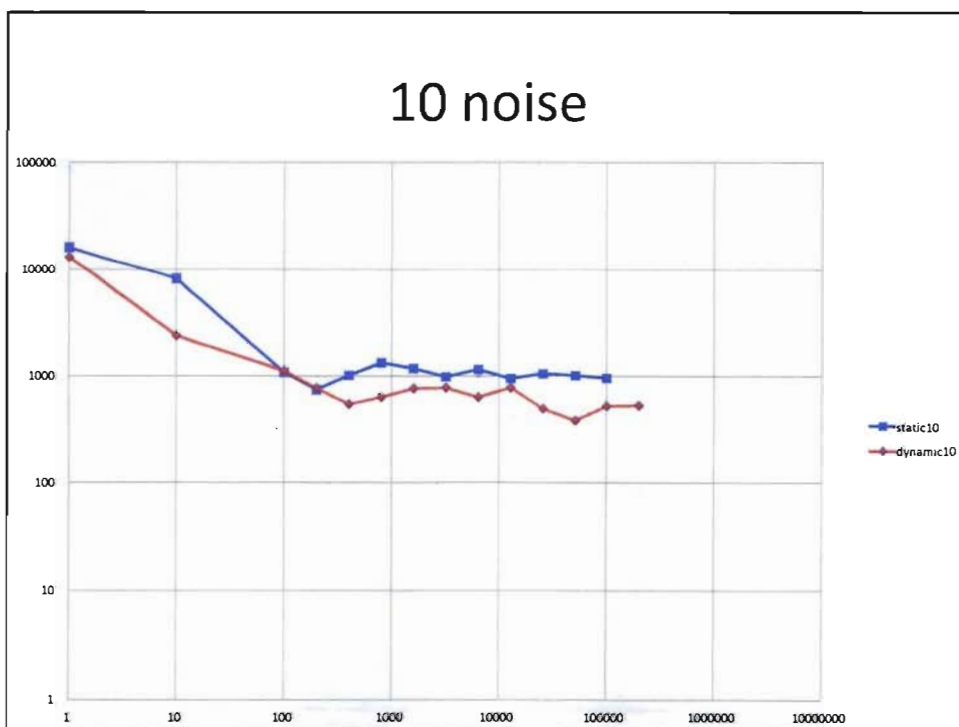
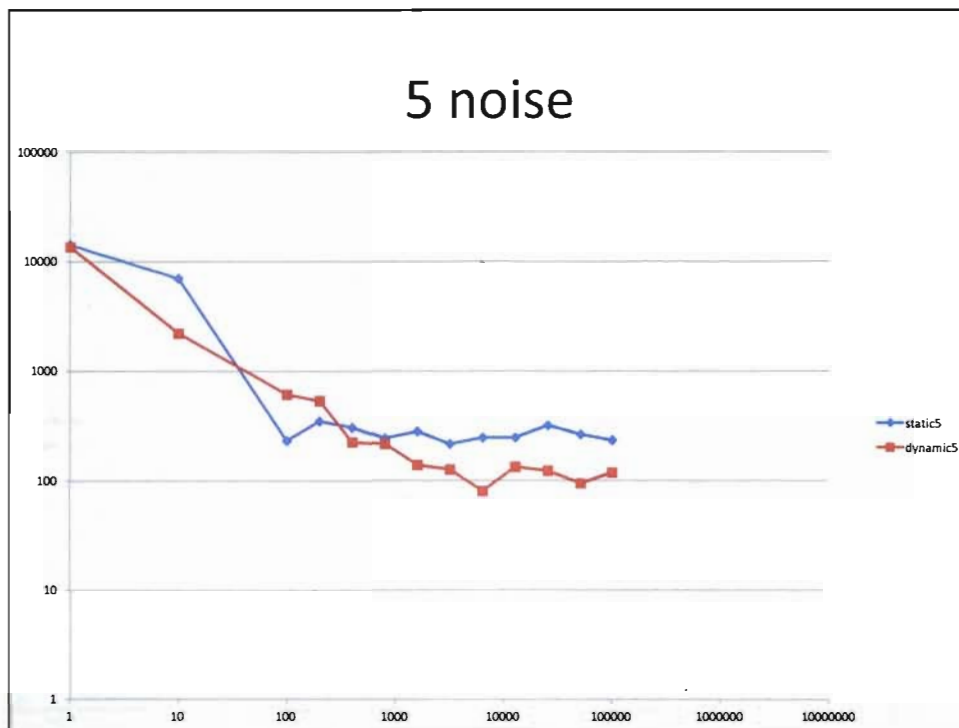
- Increased degrees of freedom will make the time series analysis more difficult



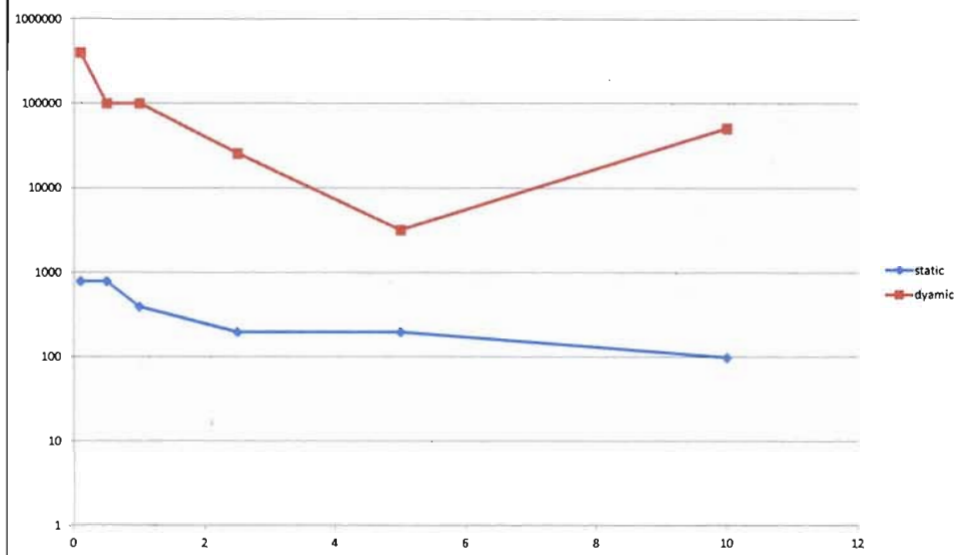




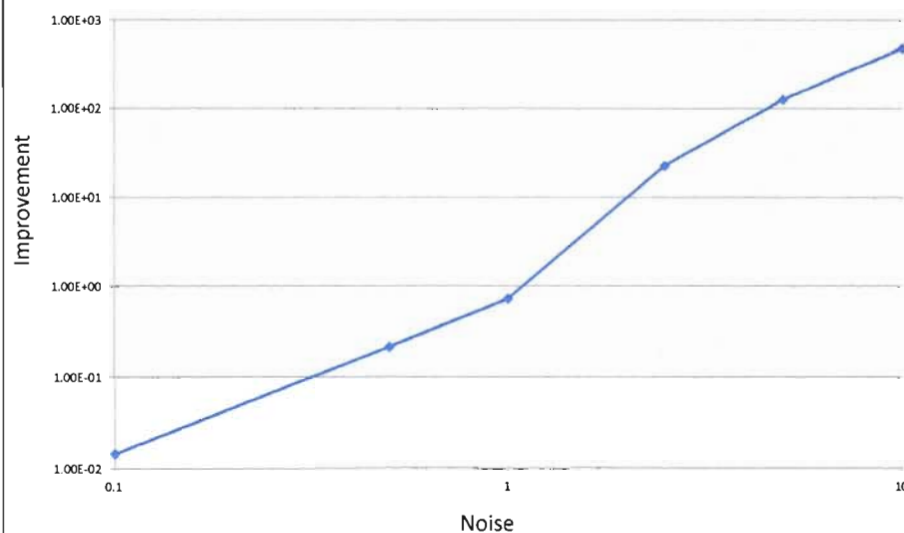


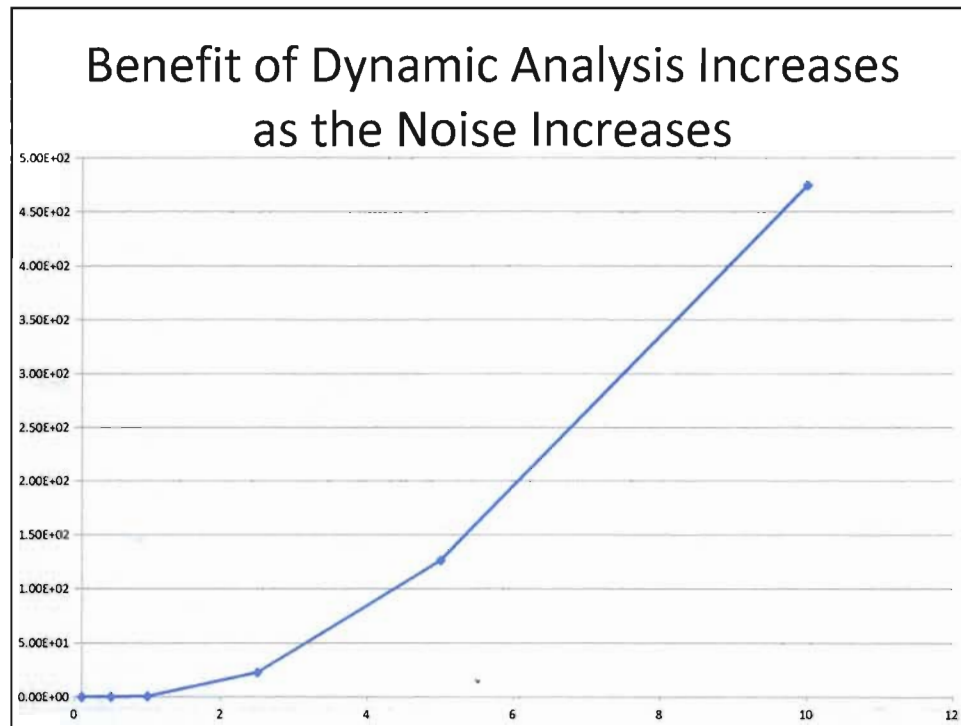


## Global Simultaneous Dynamic Optimization is Harder than Static Optimization

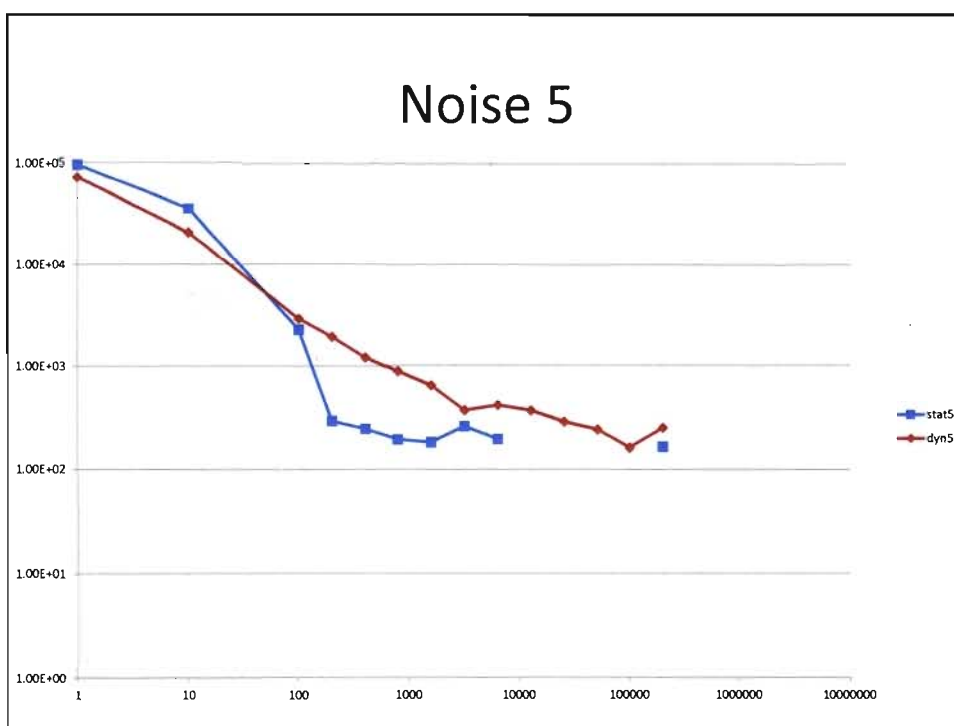
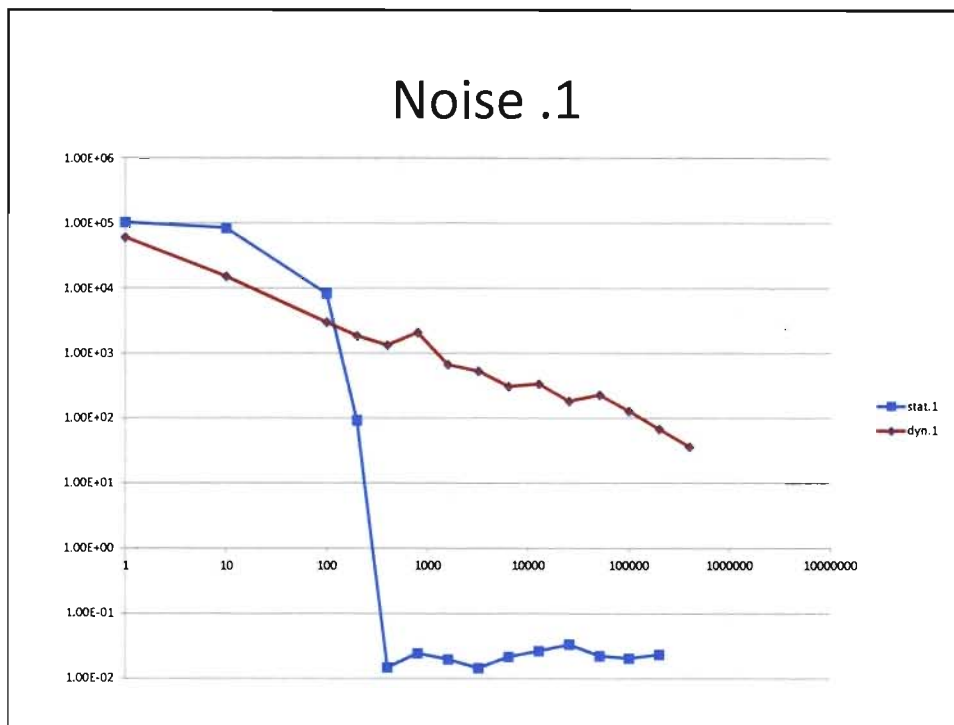


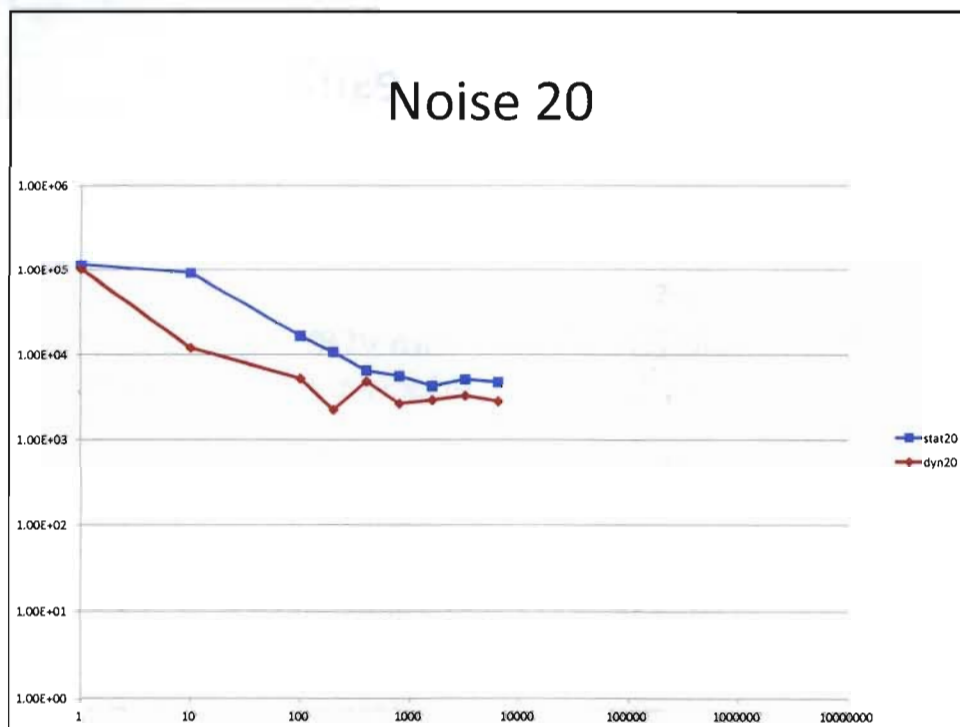
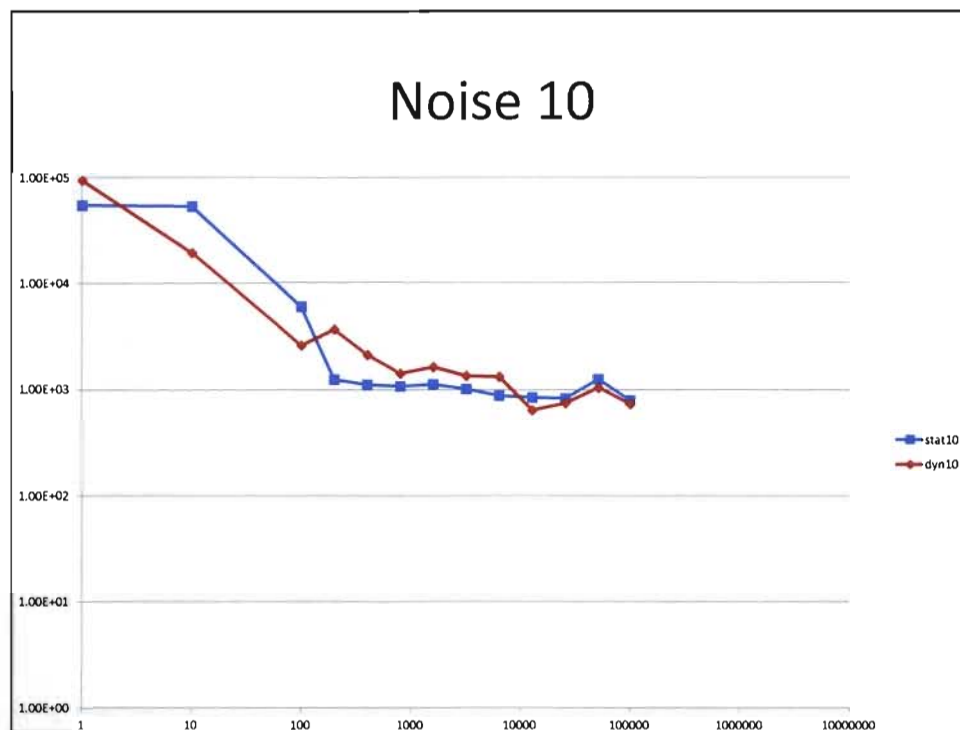
## Logarithmic 2 DF Summary Slide





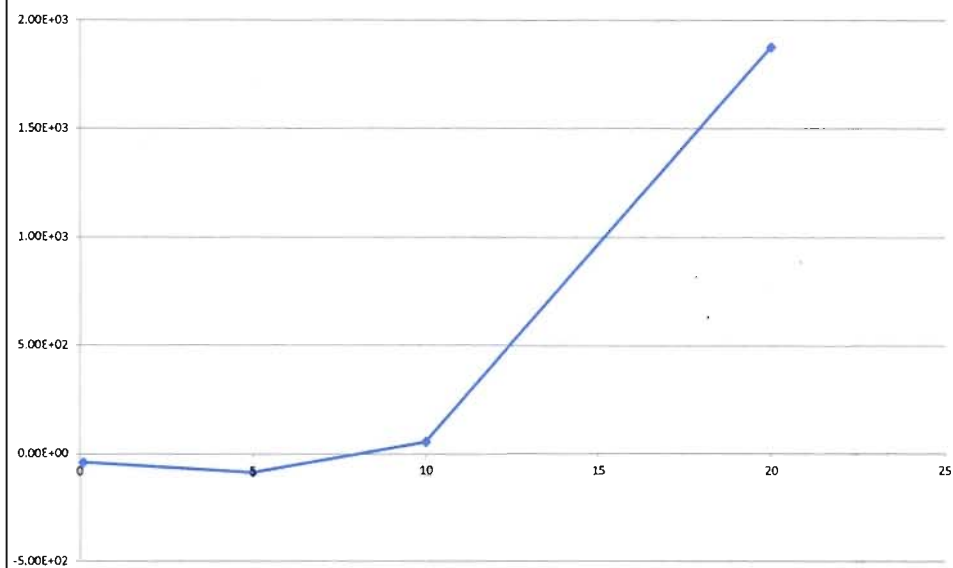
Three Degrees of Temporal  
Freedom.







## Summary, 3 Temporal Degrees of Freedom



## Time Series Results

- 1DF: Simple TSA is always superior to static analysis
- 2DF: Simple TSA is almost always superior to static analysis.
- 3DF: Simple TSA is better with VERY high Signal to Noise Ratios, 1:10 at the edges, and 15:10 in the center.
  - Average SNR?
    - $\approx 7:10$
- 4DF: Unknown...
  - Approaching an under/constrained problem

## Time Series Results

- Time series analysis involves a far more difficult optimization problem than is present in static analysis.
- Interestingly enough, with lower noise levels the optimization problem is more “difficult” in the sense that it is possible to refine the answer to a greater degree
- When the optimization problem can be solved, time series analysis can outperform static analysis for some combinations of noise and temporal degrees of freedom.
- The number of optimization steps necessary before time series analysis outperforms static analysis depends on the noise and the temporal degrees of freedom.

## Unanswered Time Series Questions

- There exists a level of temporal degrees of freedom where static analysis is always better (must be greater than 3, we need to try higher values)?
  - What happens when the time series polynomial is under constrained (more temporal degrees of freedom than there are observation time steps)
- Does that change if you move to a more complex form of time series analysis (away from simply matching a polynomial).

## Conclusions

- The new scripting language (S-BIE) allowed for complex iterative algorithmic evaluation
  - The BIE's graphical interface was insufficient for this systematic study
- Time series analysis shows potential for real applications at DARHT
  - Improvement will depend on the noise level present in the data.
  - Improvement will depend on how tightly the physics can constrain the temporal motion of the object
  - Complex global optimization will likely require improvements in the BIE's function optimization algorithms

## Future Work

- Scripting interface:
  - Incorporation of more of the BIE's tools into the new scripting language
  - Integrating the two interfaces together in one tool
- Time series analysis:
  - Explore higher temporal degrees of freedom
  - Analysis of more complex shapes
  - Implementation of more advanced function optimization routines
  - Exploration of other techniques for taking advantage of time series data besides fitting a polynomial.