Gravitational-Wave Parameter estimation with Reduced Order Quadratures

Priscilla Canizares (IoA, Cambridge)

With Scott E. Field, Jonathan R. Gair & Manuel Tiglio

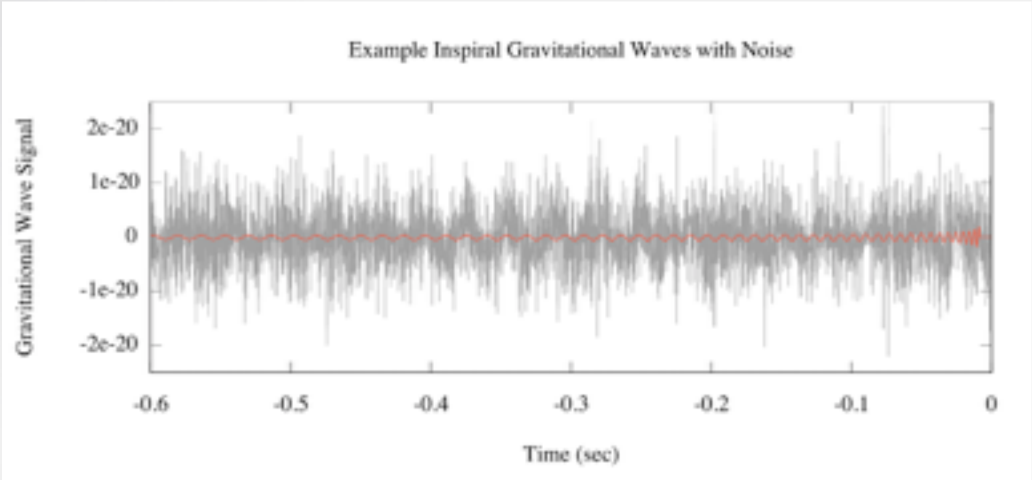
Outlook

- Introduction: Parameter estimation
- Reduced Order Modelling:
  - ☆ Reduced Basis
  - ☆ Empirical Interpolation method
  - ☆ Reduced order quadratures
- Example: sine-Gaussian burst waveforms for LIGO
- On going work: TaylorF2 waveforms
- Further applications
- Summary

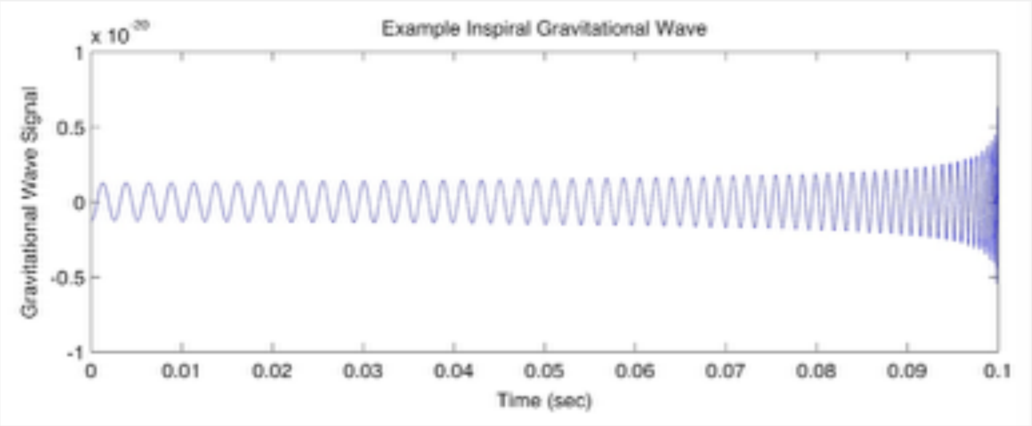
Introduction

- Accurate GW models are being developed and improved in order to dig out the GW signal $h(\lambda_{\mathbf{0}})$ from the noise n .

$$s = h(\lambda_{\mathbf{0}}) + n$$



$$h(\lambda_{\mathbf{0}})$$

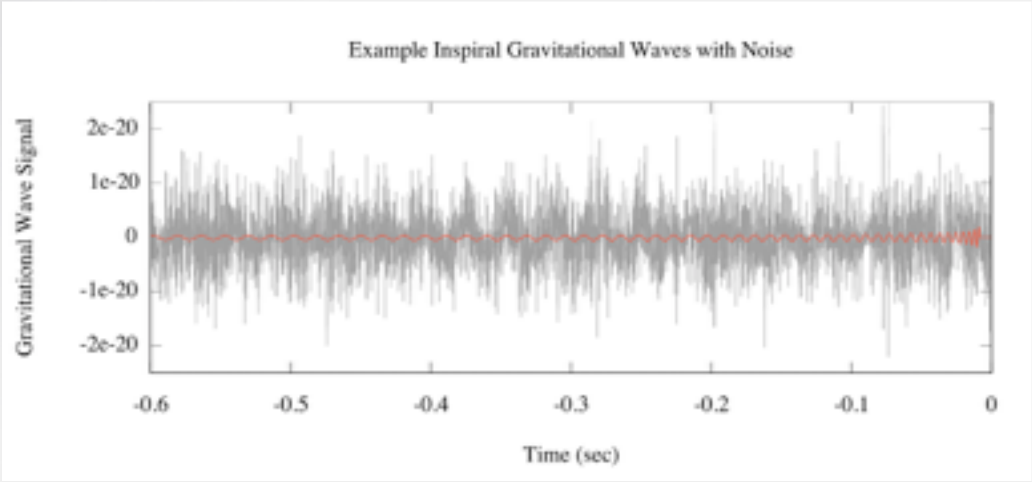


- Markov chain Monte Carlo (MCMC) or similar techniques, like nested sampling, are employed to:
  - ☆ Assess the relative likelihood of different waveform models matching the data (the likelihood of a true detection vs a false noise trigger)
  - ☆ Estimate the GW parameters by computing the **posterior distribution function** (PDF)

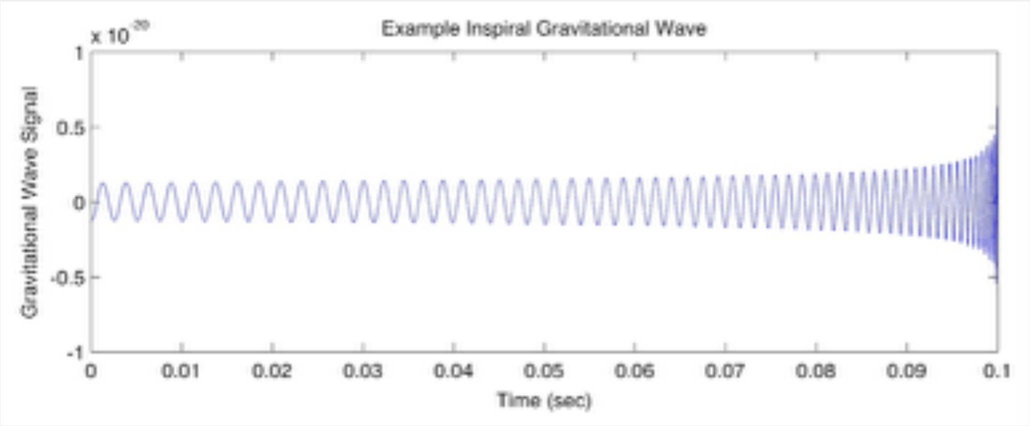
Introduction

- Accurate GW models are being developed and improved in order to dig out the GW signal $h(\lambda_\mathbf{0})$ from the noise n .

$$s = h(\lambda_\mathbf{0}) + n$$



$$h(\lambda_\mathbf{0})$$



- Markov chain Monte Carlo (MCMC) or similar techniques, like nested sampling, are employed to:
  - ☆ Assess the relative likelihood of different waveform models matching the data (the likelihood of a true detection vs a false noise trigger)
  - ☆ Estimate the GW parameters by computing the **posterior distribution function** (PDF)

Introduction

Match filtering techniques: The detector signal (GW + noise) is correlated against single GW templates over long observation times (continuous GW signals) and over a large parameter space λ_0 (searches target unknown systems).

- Parameter estimation requires repeated evaluations of the **likelihood** across the parameter space.
- MCMC techniques used to compute PDF are computationally expensive - large number of waveforms are generated and filtered against the data

$$\langle a \,|b\rangle = 4\Re \int_0^\infty \frac{\tilde{a}(f)\tilde{b}^*(f)}{\tilde{S}_n(f)} \mathrm{d}f$$

$$p(s|\lambda) = p(n = s - h(\lambda))$$

Gaussian noise $\propto \exp\left[- \langle n|n\rangle /2\right]$

$$\langle s|s\rangle + \langle h\left(\boldsymbol{\lambda}\right)|h\left(\boldsymbol{\lambda}\right)\rangle - 2\Re\langle s|h\left(\boldsymbol{\lambda}\right)\rangle$$

Introduction

Match filtering techniques: The detector signal (GW + noise) is correlated against single GW templates over long observation times (continuous GW signals) and over a large parameter space λ_0 (searches target unknown systems).

- Parameter estimation requires repeated evaluations of the **likelihood** across the parameter space.
- MCMC techniques used to compute PDF are computationally expensive - large number of waveforms are generated and filtered against the data

$$\langle a | b \rangle = 4\Re \int_0^\infty \frac{\tilde{a}(f)\tilde{b}^*(f)}{\tilde{S}_n(f)} \mathrm{d}f$$

$$p(s|\lambda) = p(n = s - h(\lambda))$$

Gaussian noise $\propto \exp\left[-\langle n|n\rangle/2\right]$

$$\langle s|s\rangle + \langle h(\boldsymbol{\lambda})|h(\boldsymbol{\lambda})\rangle - 2\Re\langle s|h(\boldsymbol{\lambda})\rangle$$

- The number of correlations/templates grows with the number of physical parameters and the number of cycles ~ $(N_{cycles})^{Npar}$ ex. $(10^5)^8 = 10^{40}$ templates for a fully coherent search.

Introduction

Match filtering techniques: The detector signal (GW + noise) is correlated against single GW templates over long observation times (continuous GW signals) and over a large parameter space λ_0 (searches target unknown systems).

- Parameter estimation requires repeated evaluations of the **likelihood** across the parameter space.
- MCMC techniques used to compute PDF are computationally expensive - large number of waveforms are generated and filtered against the data

$$\langle a | b \rangle = 4\Re \int_0^\infty \frac{\tilde{a}(f)\tilde{b}^*(f)}{\tilde{S}_n(f)} \mathrm{d}f$$

$$\begin{aligned} p(s|\lambda) &= p(n = s - h(\lambda)) \\ &\propto \exp\left[- \langle n|n\rangle /2\right] \end{aligned}$$

Gaussian noise

$$\langle s|s\rangle + \langle h(\boldsymbol{\lambda})|h(\boldsymbol{\lambda})\rangle - 2\Re\langle s|h(\boldsymbol{\lambda})\rangle$$

- The number of correlations/templates grows with the number of physical parameters and the number of cycles ~ $(N_{cycles})^{Npar}$ ex. $(10^5)^8 = 10^{40}$ templates for a fully coherent search.
- Correlation cost scale with the length of the data N, which in turn depends on both the observation time, Tobs, and sampling rate N.

Introduction

Match filtering techniques: The detector signal (GW + noise) is correlated against single GW templates over long observation times (continuous GW signals) and over a large parameter space λ_0 (searches target unknown systems).

- Parameter estimation requires repeated evaluations of the **likelihood** across the parameter space.
- MCMC techniques used to compute PDF are computationally expensive - large number of waveforms are generated and filtered against the data

$$\langle a\,|b\rangle = 4\Re\int_0^\infty \frac{\tilde{a}(f)\tilde{b}^*(f)}{\tilde{S}_n(f)}\mathrm{d}f$$

$$\begin{aligned} p(s|\lambda) &= p(n = s - h(\lambda)) \\ &\propto \exp\left[-\langle n|n\rangle/2\right] \end{aligned}$$

Gaussian noise

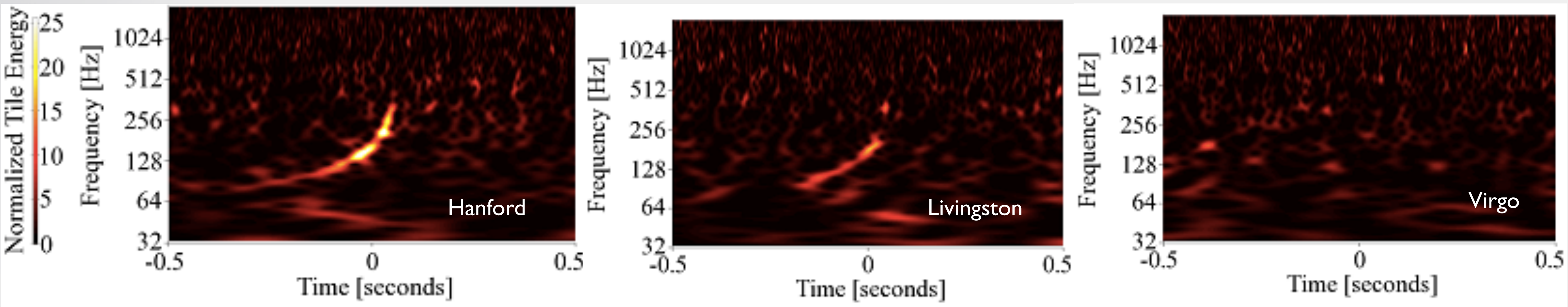
$$\langle s|s\rangle + \langle h(\boldsymbol{\lambda})\,|h(\boldsymbol{\lambda})\rangle - 2\Re\langle s|h(\boldsymbol{\lambda})\rangle$$

- The number of correlations/templates grows with the number of physical parameters and the number of cycles ~ $(N_{cycles})^{Npar}$ ex. $(10^5)^8 = 10^{40}$ templates for a fully coherent search.
- Correlation cost scale with the length of the data N, which in turn depends on both the observation time, Tobs, and sampling rate N.
- Usually the presence of noise reduce the convergence of numerical integrations.

Introduction

Example GW100916:

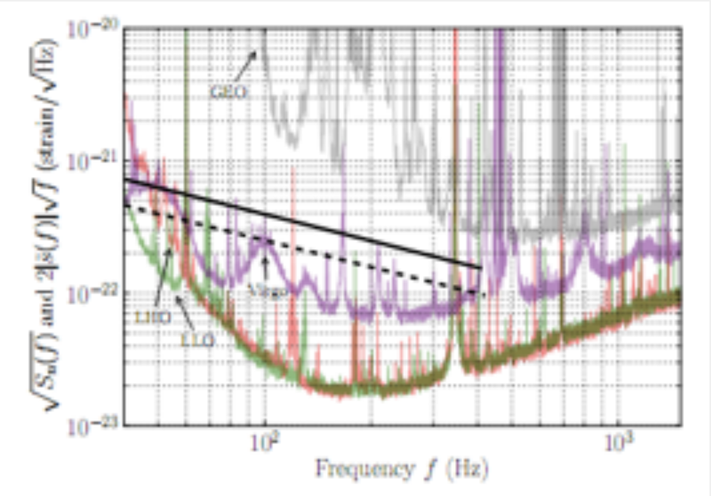
- Blind injection designed to test the analysis pipelines.



arxiv:1304.1775

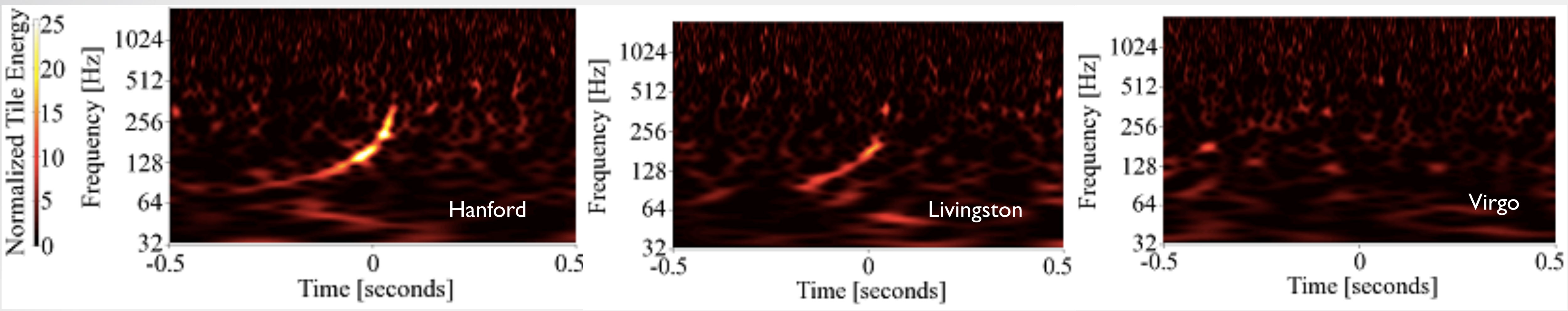
- Parameter estimation was performed using multiple algorithms and multiple waveform families - took several months!
- In aLIGO era, expect/hope for ~10s of signals a year - need faster follow-up.
- Various methods - SVD of waveform space with interpolation, neural network training or **reduced order quadrature (ROQ).**

Introduction



Example GW100916:

•Blind injection designed to test the analysis pipelines.



arxiv:1304.1775

•Parameter estimation was performed using multiple algorithms and multiple waveform families - took several months!

•In aLIGO era, expect/hope for ~10s of signals a year - need faster follow-up.

•Various methods - SVD of waveform space with interpolation, neural network training or **reduced order quadrature (ROQ).**

Reduced Order Modelling

- What we want: Compression of the data without loss of information - less computational operations.

- Usually standard interpolation is employed to represent the GW with a reduced amount of points

Reduced Order Modelling

- What we want: Compression of the data without loss of information - less computational operations.

- Usually standard interpolation is employed to represent the GW with a reduced amount of points

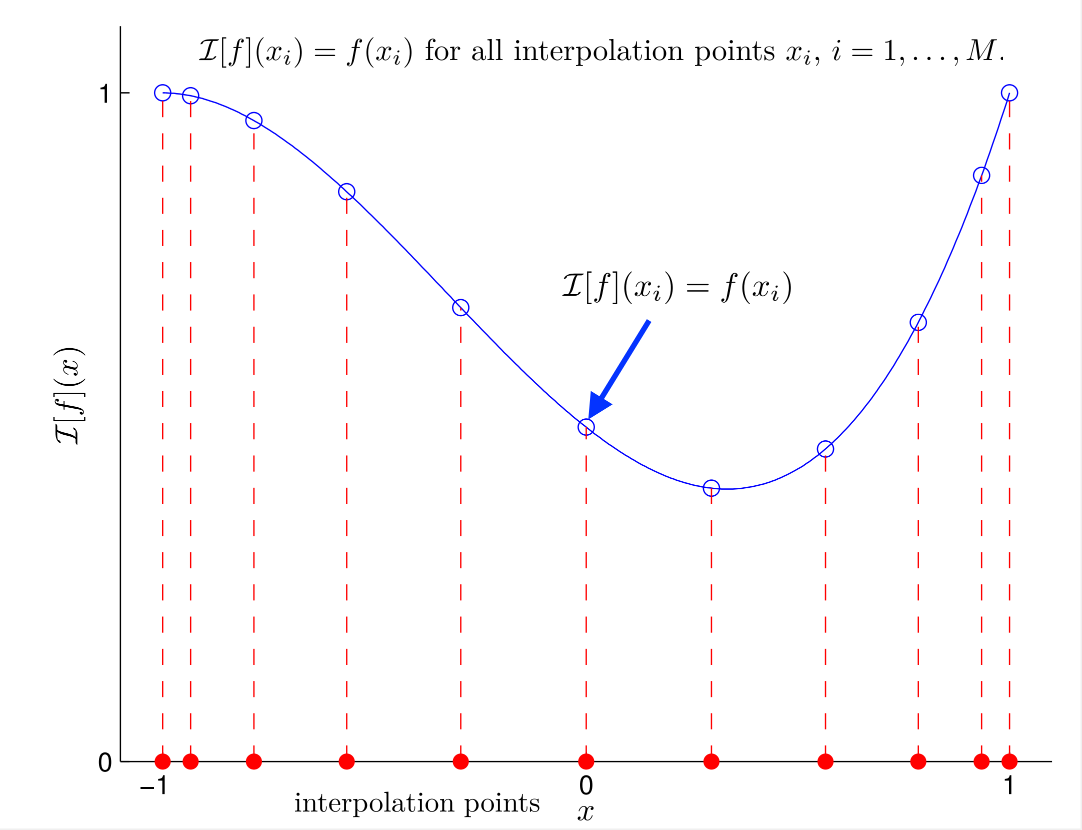
Interpolation is used to approximate a function f on a domain $\Omega,\ f:\Omega\rightarrow\mathbb{R}$ by requiring the approximation, $\mathcal{I}_M[f]:\Omega\rightarrow\mathbb{R}$, to be exact in a set of M interpolation points $\{\boldsymbol{x}_i\}_{i=1}^{M}, \boldsymbol{x}_i\in\Omega$

$$f(\boldsymbol{x}_i)=\mathcal{I}_M[f](\boldsymbol{x}_i) \qquad i=1,\ldots,M.$$

Approximates the function f by finite sums of well chosen, **pre-defined** basis functions q_i,

$$f(\boldsymbol{x})\approx\mathcal{I}_M[f](\boldsymbol{x})=\sum_{i=1}^{M}\beta_i q_i(\boldsymbol{x}).$$

Need to find the points **x**i and basis functions q_i, to minimise the approximation error.



Reduced Order Modelling

- What we want: Compression of the data without loss of information - less computational operations.

- Usually standard interpolation is employed to represent the GW with a reduced amount of points

Interpolation is used to approximate a function f on a domain $\Omega,\ f:\Omega\to\mathbb{R}$ by requiring the approximation, $\mathcal{I}_M[f]:\Omega\to\mathbb{R}$, to be exact in a set of M interpolation points $\{\boldsymbol{x}_i\}_{i=1}^M, \boldsymbol{x}_i\in\Omega$

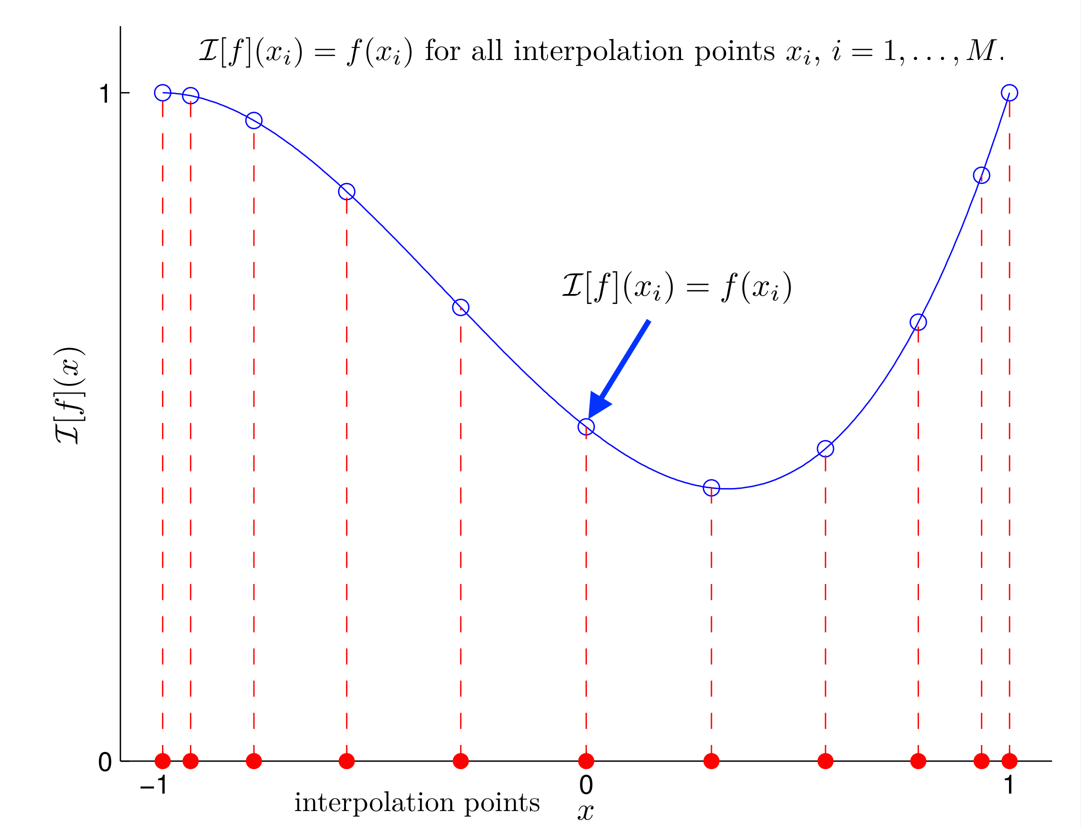
$$f(\boldsymbol{x}_i)=\mathcal{I}_M[f](\boldsymbol{x}_i) \qquad i=1,\ldots,M.$$

Approximates the function f by finite sums of well chosen, **pre-defined** basis functions q_i,

$$f(\boldsymbol{x})\approx\mathcal{I}_M[f](\boldsymbol{x})=\sum_{i=1}^M\beta_i q_i(\boldsymbol{x}).$$

Need to find the points **x**i and basis functions q_i, to minimise the approximation error.

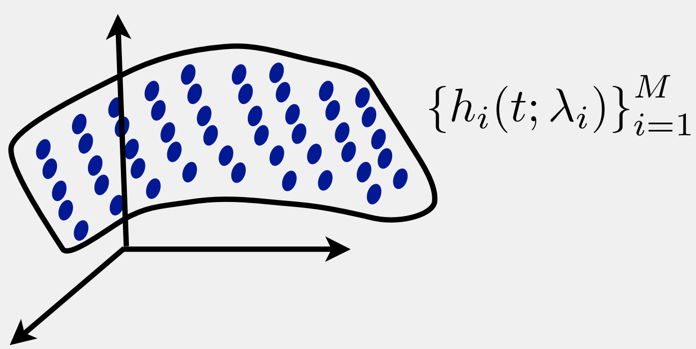
Polynomial basis functions, ex. Gauss-Lobato Legendre (GLL), yields exponential convergence for analytical functions well approximated by polynomials, and sampled at GLL points. Not true in GW data analysis applications.



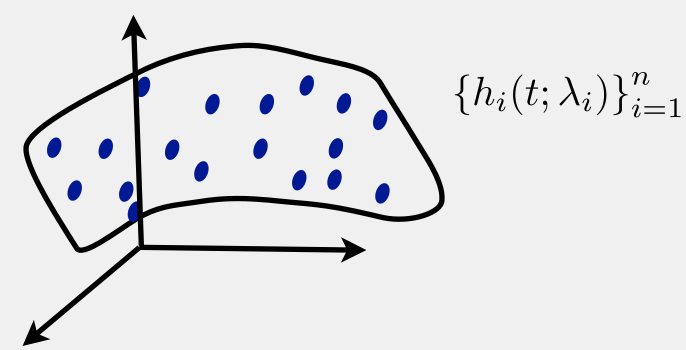
Reduced Order Modelling

Reduced Order Modelling

- **Reduced basis** (RBs) framework for modelling space: Efficiently deals with parametrised problems, in our case given by GW waveforms



Training space (template bank) covers a given range of parameters



The GWs of the training space can be represented with a basis of waveform templates - the RBs.

Reduced Basis

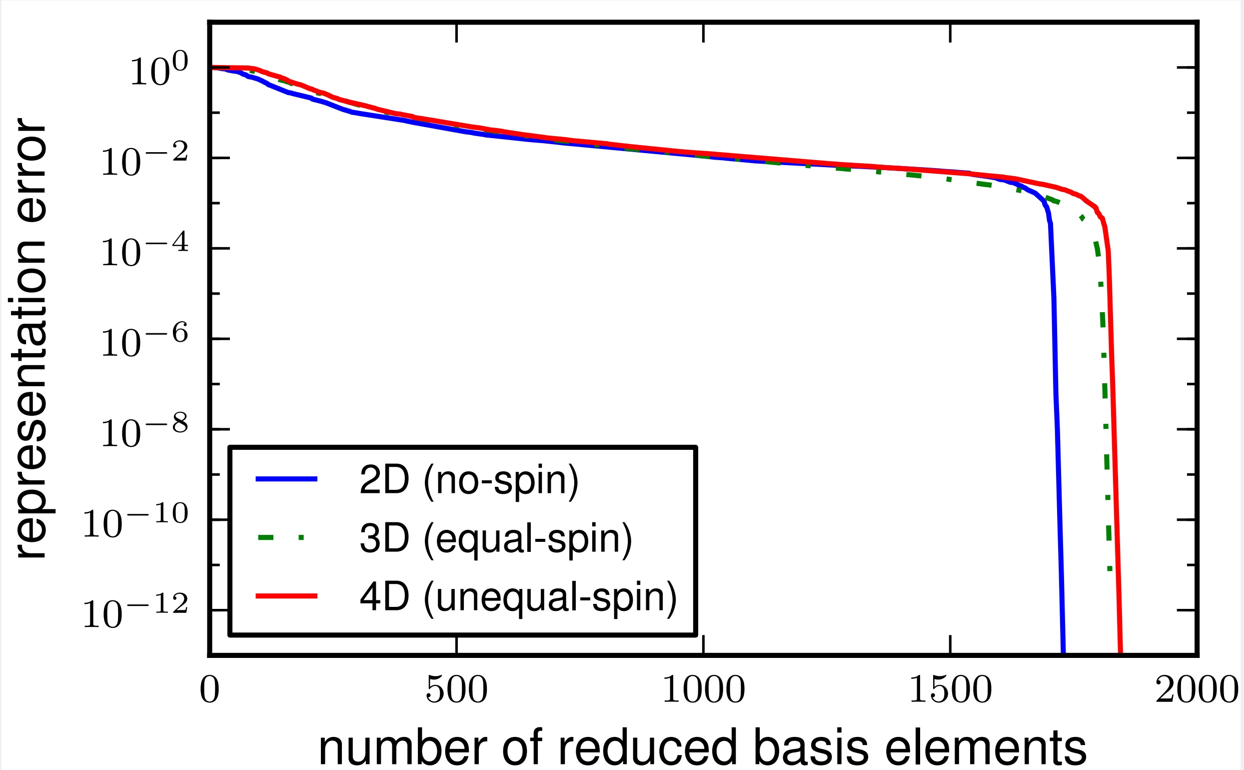
Ex. Non-spinning post-Newtonian (PN) waveforms (know in closed form) used for searches of inspirals.

Number of RB is significantly smaller

Detector	Overlap Error	BBH RB	BBH TM	BNS RB	BNS TM
LIGO	10^{-2}	165	2,450	898	10,028
	10^{-5}	170	1.2×10^{6}	904	4.3×10^{6}
	2.5×10^{-13}	182	5.9×10^{12}	917	1.4×10^{13}
AdvLIGO	10^{-2}	1,058	19,336	5,395	72,790
	10^{-5}	1,687	1.5×10^{7}	8,958	4.9×10^{7}
	2.5×10^{-13}	1,700	2.3×10^{14}	8,976	5.6×10^{14}
AdvVirgo	10^{-2}	1,395	42,496	7,482	156,127
	10^{-5}	1,690	3.1×10^{7}	8,960	8.3×10^{7}
	2.5×10^{-13}	1,703	4.8×10^{14}	8,977	6.0×10^{14}

BNS: Binary Neutron Stars , BBH: Binary Black Holes
TM = Template Metric (standard approach)

Representation error converges exponentially with the size of basis



The number of RB marginally increases when adding new parameters

Field et al. PRL 2012

Reduced Basis

Offline

1) Greedy algorithm to build a RB

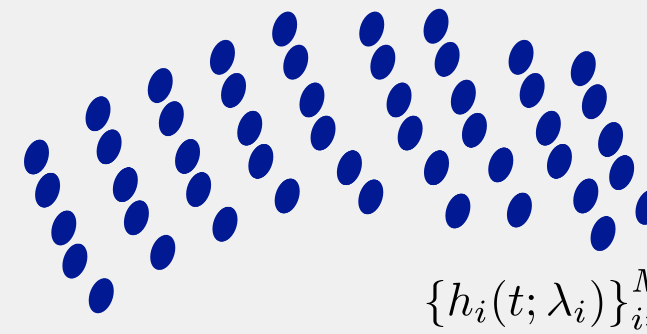
- Start with a set of training templates $\{h_i(f;\lambda_i)\}_{i=1}^{M}$ evaluated at sample of (training) points $\{\lambda_i\}_{i=1}^{M}$.

  Output $\{e_i\}_{i=1}^{m}$ the RB basis $\{\lambda_i\}_{i=1}^{m}$ and the associated points $m \ll M$.

- The first basis template is chosen at random for some j $e_1 = h_i(f;\lambda_i)$

  - Iterate: For j=2 to m

    - Compute $\mathcal{H}_k^{j-1} \equiv P_{j-1}[h_k(f;\lambda_k)]$, the projection of the remaining waveforms in the training set into the current reduced basis.



$\{h_i(t;\lambda_i)\}_{i=1}^{M}$

    - Find $K = \mathrm{argmax}||\mathcal{H}_k^{j-1} - h_k(f;\lambda_k)||^2$, the template in the training set with the largest (pointwise) representation error.

    - Set $e_j(t) = h_K(f;\lambda_K)$ and orthonormalize

    - Increment j and repeat.

- Stop when maximum representation error is less than a threshold

$$|h(\lambda) - h(\lambda)_{RB}| = |h(\lambda) - \sum_{i=1}^{m} c(\lambda_i)e_i| \ \leq \epsilon$$

Reduced Basis

1) Greedy algorithm to build a RB

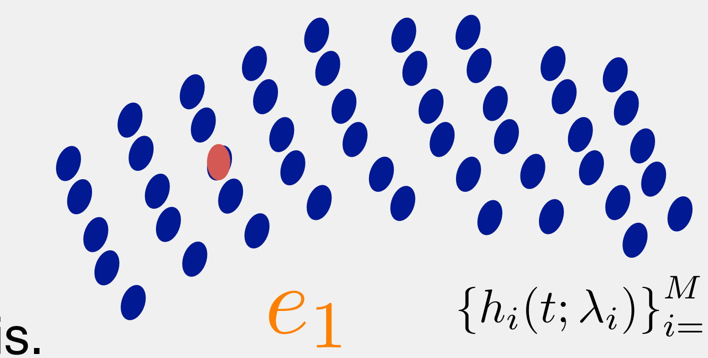
Offline

- Start with a set of training templates $\{h_i(f;\lambda_i)\}_{i=1}^{M}$ evaluated at sample of (training) points $\{\lambda_i\}_{i=1}^{M}$.

  Output $\{e_i\}_{i=1}^{m}$ the RB basis $\{\lambda_i\}_{i=1}^{m}$ and the associated points $m \ll M$.

- The first basis template is chosen at random for some j $e_1 = h_j(f;\lambda_j)$

  - Iterate: For j=2 to m



    - Compute $\mathcal{H}_k^{j-1} \equiv P_{j-1}[h_k(f;\lambda_k)]$, the projection of the remaining waveforms in the training set into the current reduced basis.
    - Find $K = \mathrm{argmax}||\mathcal{H}_k^{j-1} - h_k(f;\lambda_k)||^2$, the template in the training set with the largest (pointwise) representation error.
    - Set $e_j(t) = h_K(f;\lambda_K)$ and orthonormalize
    - Increment j and repeat.

- Stop when maximum representation error is less than a threshold

$$|h(\lambda) - h(\lambda)_{RB}| = |h(\lambda) - \sum_{i=1}^{m} c(\lambda_i)e_i| \leq \epsilon$$

Reduced Basis

Offline

1) Greedy algorithm to build a RB

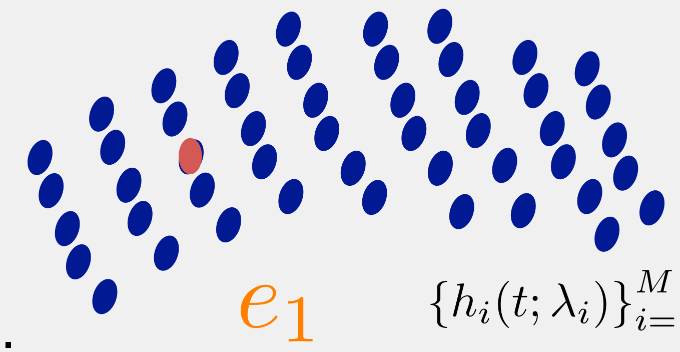
- Start with a set of training templates $\{h_i(f;\lambda_i)\}_{i=1}^{M}$ evaluated at sample of (training) points $\{\lambda_i\}_{i=1}^{M}$.

  Output $\{e_i\}_{i=1}^{m}$ the RB basis $\{\lambda_i\}_{i=1}^{m}$ and the associated points $m \ll M$.

- The first basis template is chosen at random for some j $e_1 = h_i(f;\lambda_i)$

  - Iterate: For j=2 to m

    - Compute $\mathcal{H}_k^{j-1} \equiv P_{j-1}[h_k(f;\lambda_k)]$, the projection of the remaining waveforms in the training set into the current reduced basis.



    - Find $K = \text{argmax}||\mathcal{H}_k^{j-1} - h_k(f;\lambda_k)||^2$, the template in the training set with the largest (pointwise) representation error.

    - Set $e_j(t) = h_K(f;\lambda_K)$ and orthonormalize

    - Increment j and repeat.

- Stop when maximum representation error is less than a threshold

$$|h(\lambda) - h(\lambda)_{RB}| = |h(\lambda) - \sum_{i=1}^{m} c(\lambda_i) e_i| \leq \epsilon$$

Reduced Basis

Offline

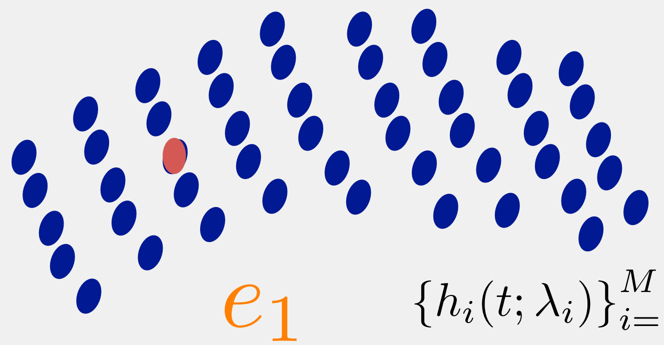
1) Greedy algorithm to build a RB

- Start with a set of training templates $\{h_i(f;\lambda_i)\}_{i=1}^{M}$ evaluated at sample of (training) points $\{\lambda_i\}_{i=1}^{M}$.

  Output $\{e_i\}_{i=1}^{m}$ the RB basis $\{\lambda_i\}_{i=1}^{m}$ and the associated points $m \ll M$.

- The first basis template is chosen at random for some j $e_1 = h_i(f;\lambda_i)$

  - Iterate: For j=2 to m



    - Compute $\mathcal{H}_k^{j-1} \equiv P_{j-1}[h_k(f;\lambda_k)]$, the projection of the remaining waveforms in the training set into the current reduced basis.
    - Find $K = \mathrm{argmax}||\mathcal{H}_k^{j-1} - h_k(f;\lambda_k)||^2$, the template in the training set with the largest (pointwise) representation error.
    - Set $e_j(t) = h_K(f;\lambda_K)$ and orthonormalize
    - Increment j and repeat.

- Stop when maximum representation error is less than a threshold

$$|h(\lambda) - h(\lambda)_{RB}| = |h(\lambda) - \sum_{i=1}^{m} c(\lambda_i)e_i| \leq \epsilon$$

Reduced Basis

1) Greedy algorithm to build a RB

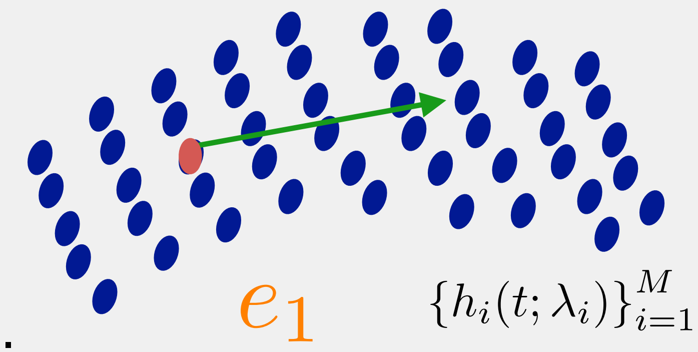
Offline

☆ Start with a set of training templates $\{h_i(f;\lambda_i)\}_{i=1}^{M}$ evaluated at sample of (training) points $\{\lambda_i\}_{i=1}^{M}$.

Output $\{e_i\}_{i=1}^{m}$ the RB basis $\{\lambda_i\}_{i=1}^{m}$ and the associated points $m \ll M$.

☆ The first basis template is chosen at random for some j $e_1 = h_i(f;\lambda_i)$

☆ Iterate: For j=2 to m



- Compute $\mathcal{H}_k^{j-1} \equiv P_{j-1}[h_k(f;\lambda_k)]$, the projection of the remaining waveforms in the training set into the current reduced basis.
- Find $K = \text{argmax}||\mathcal{H}_k^{j-1} - h_k(f;\lambda_k)||^2$, the template in the training set with the largest (pointwise) representation error.
- Set $e_j(t) = h_K(f;\lambda_K)$ and orthonormalize
- Increment j and repeat.

☆ Stop when maximum representation error is less than a threshold

$$|h(\lambda) - h(\lambda)_{RB}| = |h(\lambda) - \sum_{i=1}^{m} c(\lambda_i)e_i| \leq \epsilon$$

Reduced Basis

Offline

1) Greedy algorithm to build a RB

- Start with a set of training templates $\{h_i(f;\lambda_i)\}_{i=1}^{M}$ evaluated at sample of (training) points $\{\lambda_i\}_{i=1}^{M}$.

  Output $\{e_i\}_{i=1}^{m}$ the RB basis $\{\lambda_i\}_{i=1}^{m}$ and the associated points $m \ll M$.

- The first basis template is chosen at random for some j $e_1 = h_i(f;\lambda_i)$

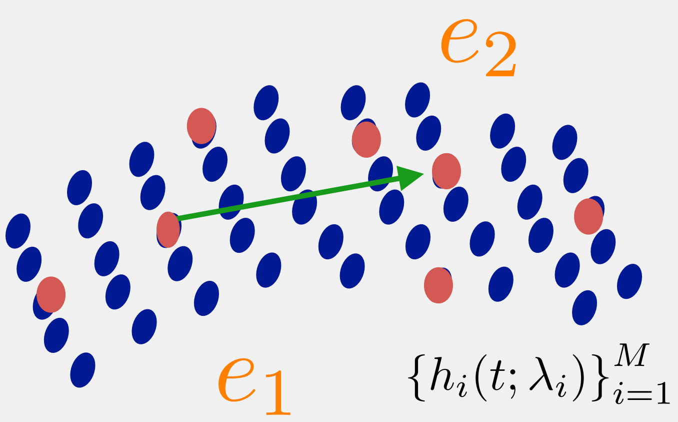
  - Iterate: For j=2 to m

    - Compute $\mathcal{H}_k^{j-1} \equiv P_{j-1}[h_k(f;\lambda_k)]$, the projection of the remaining waveforms in the training set into the current reduced basis.

    - Find $K = \mathrm{argmax}||\mathcal{H}_k^{j-1} - h_k(f;\lambda_k)||^2$, the template in the training set with the largest (pointwise) representation error.

    - Set $e_j(t) = h_K(f;\lambda_K)$ and orthonormalize

    - Increment j and repeat.



- Stop when maximum representation error is less than a threshold

$$|h(\lambda) - h(\lambda)_{RB}| = |h(\lambda) - \sum_{i=1}^{m} c(\lambda_i)e_i| \leq \epsilon$$

Reduced Basis

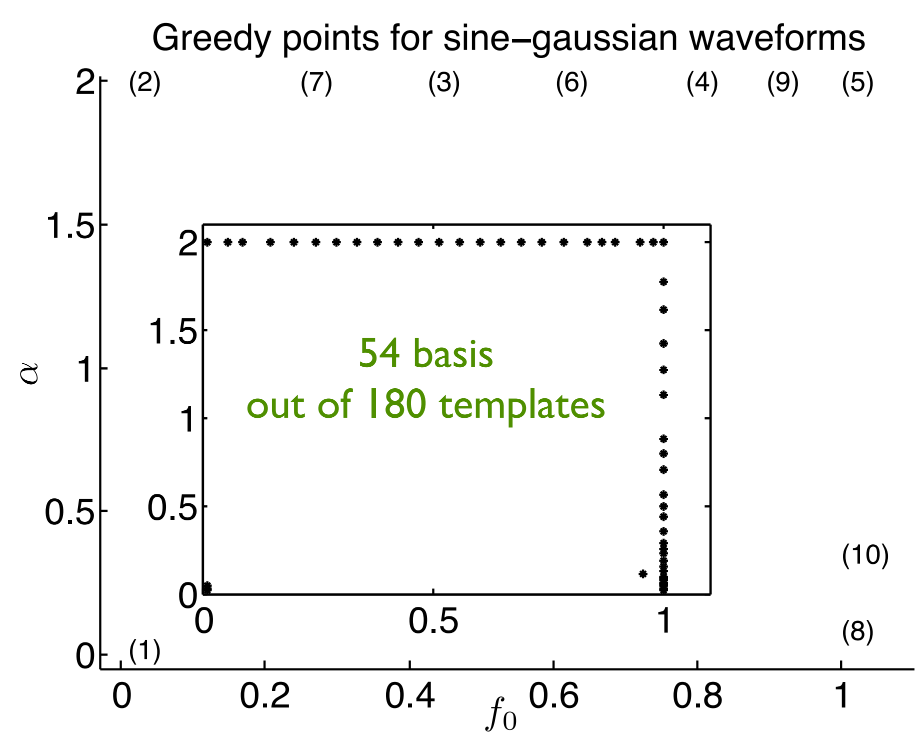
Example: Sine-Gaussian burst waveform

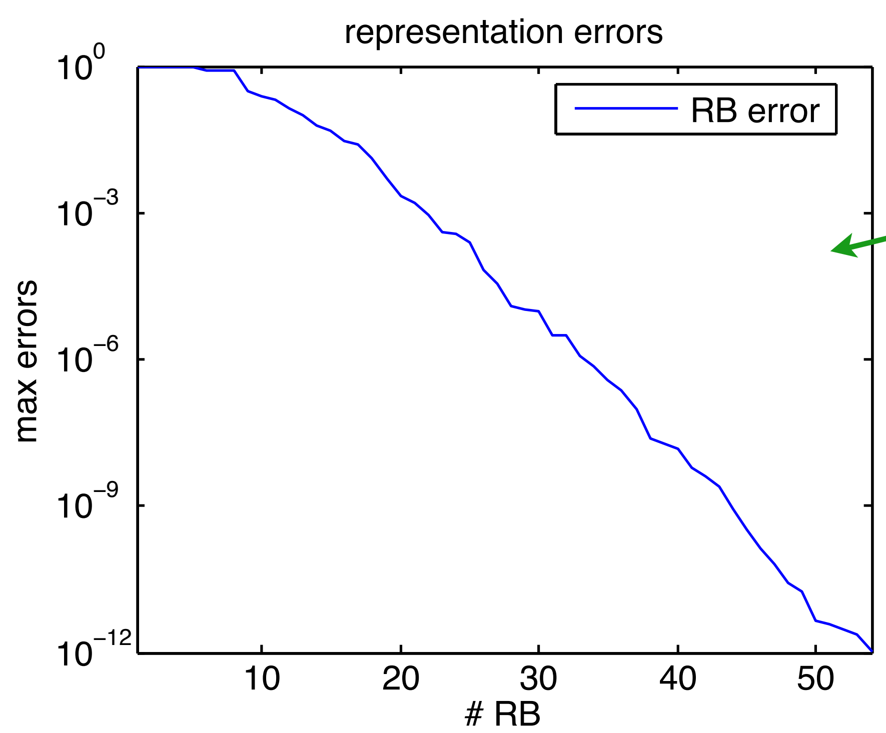
Sine-Gaussian burst model used to represent generic burst source, e.g. supernova. The waveforms are described by four parameters A, α, f_0, t_c

$$\widetilde{h}(f, \lambda) = i2\sqrt{2\pi}\alpha e^{(i2\pi t_c - 2\pi^2\alpha^2(f_0^2+f^2))} \sinh\left(4\pi^2\alpha^2 f_0 f\right)$$

•The most important ones are the burst width and frequency.

$$\sigma_m := \max_{\boldsymbol{\lambda}} \min_{c_i \in \mathbb{C}} \left\| h(\cdot;\boldsymbol{\lambda}) - \sum_{i=1}^{m} c_i(\boldsymbol{\lambda}) e_i(\cdot) \right\|^2 \leq \epsilon$$





Canizares et al PRD 2013

Reduced Basis

Caveats

- Not all linear combinations of basis templates represent physical waveforms.
- Must restrict evaluation of likelihood/posterior to physically reasonable combinations.
- On the fly projection of each waveform onto basis is expensive - lose savings.
- Interpolation between points in parameter space can be used, but only in small parameter regions.

Empirical Interpolation

Offline

- **Empirical interpolation method** (EIM) employed to approximates parametrised functions $h(\mu, \lambda)$ as

$$h(\mu, \lambda) \approx \sum_{i=1}^{M} c_i(\lambda) e_i(\mu) \qquad \text{where } \mu \text{ is } f \text{ or } t$$

The EIM is an algorithm for constructing interpolation points for a given set of basis functions iteratively,

Empirical Interpolation

Offline

2) Empirical interpolation method: Find the RB evaluation points $\mu = f$ or t

- EIM is deals with parametrised problems characterised by non-polynomial bases.
- The set of EIM points is nested and hierarchical,
- Easily handles unstructured meshes in several dimensions.

Empirical Interpolation

Offline

2) Empirical interpolation method: Find the RB evaluation points $\mu = f$ or t

- EIM is deals with parametrised problems characterised by non-polynomial bases.
- The set of EIM points is nested and hierarchical,
- Easily handles unstructured meshes in several dimensions.

We impose that the value of the RB-waveform at a given (interpolation) points is the same that the one obtained by evaluating the true signal

Empirical Interpolation

Offline

2) Empirical interpolation method: Find the RB evaluation points μ = f or t

- ☆ EIM is deals with parametrised problems characterised by non-polynomial bases.
- ☆ The set of EIM points is nested and hierarchical,
- ☆ Easily handles unstructured meshes in several dimensions.

We impose that the value of the RB-waveform at a given (interpolation) points is the same that the one obtained by evaluating the true signal

Input the set of basis functions $\{e_i\}_{i=1}^{m}$ and points $\{f\}_{i=1}^{N}$ Output set of m EIM points $\{F_i\}_{i=1}^{m} \subset \{f_i\}_{i=1}^{N}$ where the corresponding base elements are evaluated.

$$\mathcal{I}_m[h](f,\lambda) \equiv \sum_{i=1}^{m} c_i e_i(f) \longrightarrow \mathcal{I}_m[h](F_i,\lambda) = h(F_i,\lambda) \qquad \{F_i\}_{i=1}^{m} \subset \{f_i\}_{i=0}^{N}$$

$$\Rightarrow \vec{c} = A^{-1}\vec{h} \qquad \text{where } A = \begin{pmatrix} e_1(F_1) & e_2(F_1) & \cdots \\ e_1(F_2) & e_2(F_2) & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix}$$ is parameter independent

Empirical Interpolation

Offline

2) Empirical interpolation method: Find the RB evaluation points $\mu = f$ or t

- EIM is deals with parametrised problems characterised by non-polynomial bases.
- The set of EIM points is nested and hierarchical,
- Easily handles unstructured meshes in several dimensions.

We impose that the value of the RB-waveform at a given (interpolation) points is the same that the one obtained by evaluating the true signal

Input the set of basis functions $\{e_i\}_{i=1}^{m}$ and points $\{f\}_{i=1}^{N}$ Output set of m EIM points $\{F_i\}_{i=1}^{m} \subset \{f_i\}_{i=1}^{N}$ where the corresponding base elements are evaluated.

$$\mathcal{I}_m[h](f,\lambda) \equiv \sum_{i=1}^{m} c_i e_i(f) \longrightarrow \mathcal{I}_m[h](F_i,\lambda) = h(F_i,\lambda) \qquad \{F_i\}_{i=1}^{m} \subset \{f_i\}_{i=0}^{N}$$

$$\Rightarrow \vec{c} = A^{-1}\vec{h} \qquad \text{where } A = \begin{pmatrix} e_1(F_1) & e_2(F_1) & \cdots \\ e_1(F_2) & e_2(F_2) & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix}$$ is parameter independent

- Need only to evaluate the GW model at this subset of $m \ll N$ points in order to compute the coefficients c_i of the reduced basis.

Empirical Interpolation

Offline

Greedy algorithm

Start with input sample points $\{f\}_{i=1}^{N}$ and basis $\{e_i\}_{i=1}^{m}$

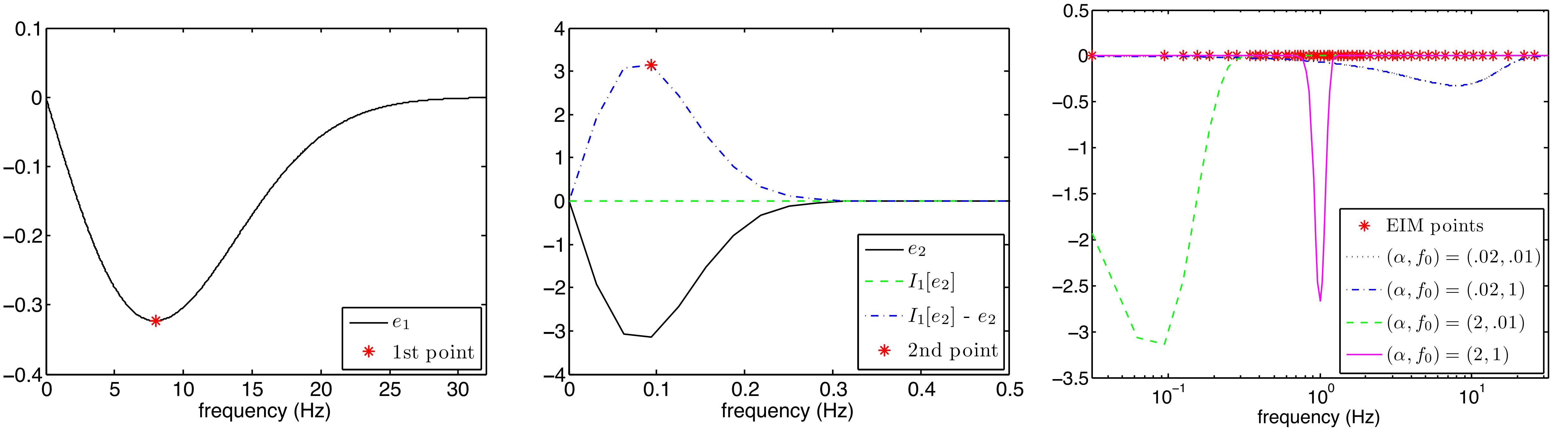
Set F_1 = argmax e_1, where argmax gives the point f_i at which e_1 has its largest value: $|e(F_1)| \geq |e_1(i)| \quad \forall i$

For j=2 to m repeat

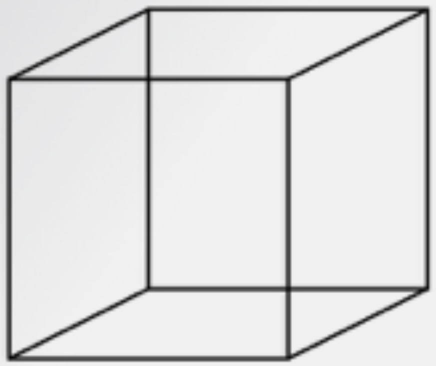
- Find $\mathcal{I}_j[e_{j+1}] = \sum_{j=1}^{m} c_j e_j(f)$ where $c_j = e_{j+i}(F_j)/e_j(F_j)$
- Set $F_{j+1} = \text{argmax}|e_j - \mathcal{I}_j[e_{j+1}]|$ (maximum pointwise error)

- Greedy algorithm ensures exponential convergence with the number of basis templates/interpolation points.

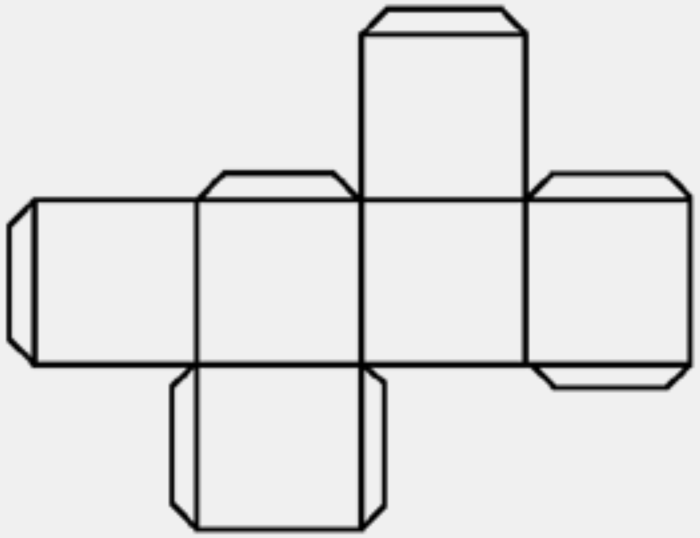
$$\max_{\boldsymbol{\lambda}} \|h(\cdot;\boldsymbol{\lambda}) - \mathcal{I}_m[h(\cdot;\boldsymbol{\lambda})]\|^2 \leq \Lambda_m^2 \sigma_m$$



Canizares et al PRD 2013



Reduced Order Quadratures



Reduced Order Quadratures

In general, if we have an interpolation $h(x) \simeq \sum_{i=1}^{m} h(x_i) l_i(x)$.

Then we can compute integrals $\int h(x)dx \simeq \sum_{i=1}^{m} \alpha_i h(x_i)$ where $\alpha_i \int l_i(x)dx$ are the weights that span the function, and provide exact integration for each ℓ_i as well as their linear combinations.

Standard examples of a quadrature rule are the familiar trapezoidal and Simpson's rules.

Reduced Order Quadratures

In general, if we have an interpolation $h(x) \simeq \sum_{i=1}^{m} h(x_i) l_i(x)$.

Then we can compute integrals $\int h(x)dx \simeq \sum_{i=1}^{m} \alpha_i h(x_i)$ where $\alpha_i \int l_i(x)dx$ are the weights that span the function, and provide exact integration for each ℓ_i as well as their linear combinations.

Standard examples of a quadrature rule are the familiar trapezoidal and Simpson's rules.

- Reduced basis representation allows efficient evaluation of quadratures (integrals).

$$\langle h(\lambda)|s\rangle \approx \sum_{i=0}^{N} \left[\frac{h(f_i;\lambda) s^*(f_i)}{S_n(f_i)} \right] \Delta f_i$$

Finds the smallest N while maintaining the accuracy needed for parameter estimation studies.

Reduced Order Quadratures

Savings in computing overlaps ~N/m

- ROQ rule

$$\langle h(\lambda)|s\rangle_d = 4\Re\int_0^\infty h(f,\lambda)s^*(f)df \;\simeq 4\Re\sum_{k=0}^{N} s^*(f_k)h(f_k;\lambda)\Delta f$$

Reduced Order Quadratures

Savings in computing overlaps ~N/m

- ROQ rule

$$\langle h(\lambda)|s\rangle_d = 4\Re \int_0^\infty h(f,\lambda)s^*(f)df \simeq 4\Re \sum_{k=0}^{N} s^*(f_k)h(f_k;\lambda)\Delta f$$

$$\simeq 4\Re \sum_{k=0}^{N} s^*(f_k)\mathcal{I}_m[h(f_k;\lambda)]\Delta f$$

$$\mathcal{I}_m[h](f,\lambda) \equiv \sum_{i=1}^{m} c_i e_i(f)$$

Reduced Order Quadratures

Savings in computing overlaps ~N/m

- ROQ rule

$$\langle h(\lambda)|s\rangle_d = 4\Re \int_0^\infty h(f,\lambda)s^*(f)df \simeq 4\Re \sum_{k=0}^{N} s^*(f_k)h(f_k;\lambda)\Delta f$$

$$\simeq 4\Re \sum_{k=0}^{N} s^*(f_k)\mathcal{I}_m[h(f_k;\lambda)]\Delta f$$

$$\mathcal{I}_m[h](f,\lambda) \equiv \sum_{i=1}^{m} c_i e_i(f)$$

$$= 4\Re \sum_{k=0}^{N} s^*(f_k)\vec{e}(f_k)A^{-1}\vec{h}(\lambda)\Delta f$$

Reduced Order Quadratures

Savings in computing overlaps ~N/m

- ROQ rule

$$\langle h(\lambda)|s\rangle_d = 4\Re \int_0^\infty h(f,\lambda)s^*(f)df \simeq 4\Re \sum_{k=0}^{N} s^*(f_k)h(f_k;\lambda)\Delta f$$

$$\simeq 4\Re \sum_{k=0}^{N} s^*(f_k)\mathcal{I}_m[h(f_k;\lambda)]\Delta f$$

$$\mathcal{I}_m[h](f,\lambda) \equiv \sum_{i=1}^{m} c_i e_i(f)$$

$$= 4\Re \sum_{k=0}^{N} s^*(f_k)\vec{e}(f_k)A^{-1}\vec{h}(\lambda)\Delta f$$

$\vec{c}$

Reduced Order Quadratures

Savings in computing overlaps ~N/m

- ROQ rule

$$\langle h(\lambda)|s\rangle_d = 4\Re\int_0^\infty h(f,\lambda)s^*(f)df \simeq 4\Re\sum_{k=0}^{N} s^*(f_k)h(f_k;\lambda)\Delta f$$

$$\simeq 4\Re\sum_{k=0}^{N} s^*(f_k)\mathcal{I}_m[h(f_k;\lambda)]\Delta f$$

$$\mathcal{I}_m[h](f,\lambda) \equiv \sum_{i=1}^{m} c_i e_i(f)$$

$$= 4\Re\sum_{k=0}^{N} s^*(f_k)\vec{e}(f_k)A^{-1}\vec{h}(\lambda)\Delta f$$

$$\vec{c}$$

$$= 4\Re\sum_{k=0}^{m} w_k h(F_k;\lambda)$$

How much $m \ll M$ is model dependent

Signal specific weights, computed once we have the data

Online

Reduced Order Quadratures

Savings in computing overlaps ~N/m

- ROQ rule

$$\langle h(\lambda)|s\rangle_d = 4\Re\int_0^\infty h(f,\lambda)s^*(f)df \simeq 4\Re\sum_{k=0}^{N} s^*(f_k)h(f_k;\lambda)\Delta f$$

$$\simeq 4\Re\sum_{k=0}^{N} s^*(f_k)\mathcal{I}_m[h(f_k;\lambda)]\Delta f$$

$$\mathcal{I}_m[h](f,\lambda) \equiv \sum_{i=1}^{m} c_i e_i(f)$$

$$= 4\Re\sum_{k=0}^{N} s^*(f_k)\vec{e}(f_k)A^{-1}\vec{h}(\lambda)\Delta f$$

$\vec{c}$

$$= 4\Re\sum_{k=0}^{m} w_k h(F_k;\lambda)$$

How much $m \ll M$ is model dependent

Signal specific weights, computed once we have the data

Online

Reduced Order Quadratures

- **Noise does not affect the computation of overlaps $\langle s|h(\boldsymbol{\lambda})\rangle$ using ROQ**

Detected signal $s(t,\lambda) = h(t,\lambda) + n(t)$

The error of the projection of the true GW with the RB one δs is exponentially small

$$h(\boldsymbol{\lambda}) = \sum_i \langle h(\boldsymbol{\lambda})|h^i_{RB}\rangle h^i_{RB} + \delta h \equiv \sum_i \alpha_i(\boldsymbol{\lambda}) h^i_{RB} + \delta h^{\perp}$$

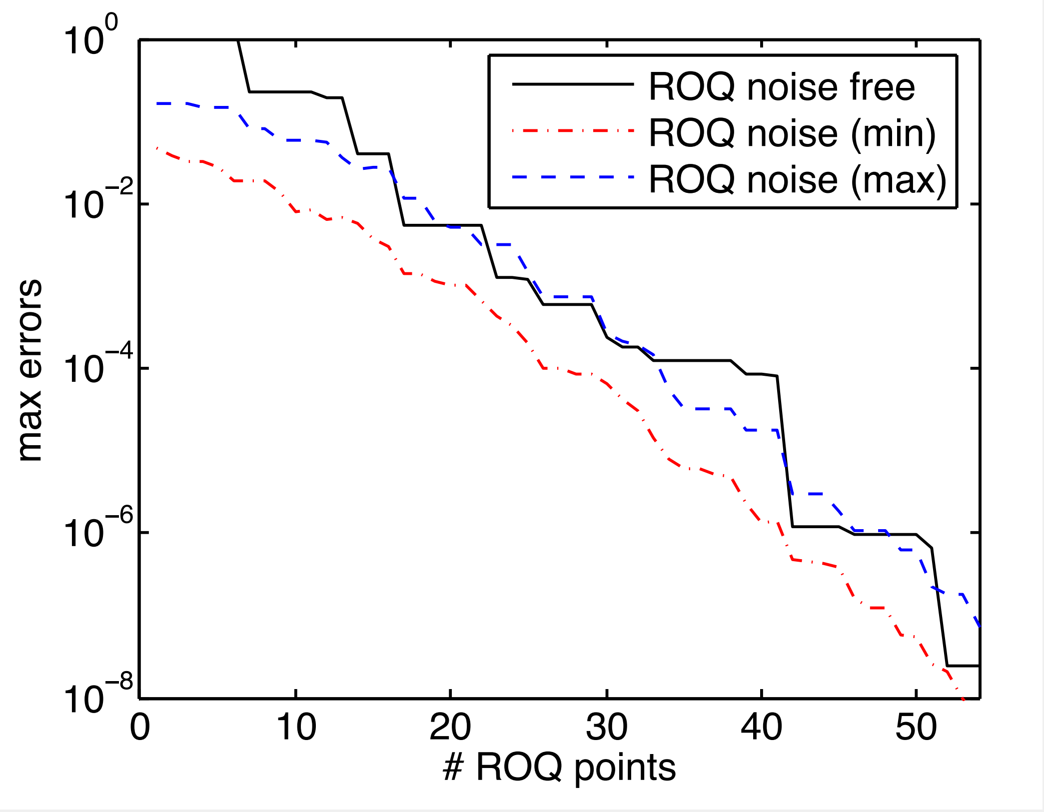
The error of the projection of the detected signal with the RB one is $\delta s \sim s$ due to the noise

$$s = \sum_i \langle s|h^i_{RB}\rangle h^i_{RB} + \delta s \equiv \sum_i \beta_i(\boldsymbol{\lambda}) h^i_{RB} + \delta s$$

cross terms cancel due to orthogonality condition

and using that δs is exponentially small

$$\langle s|h(\boldsymbol{\lambda})\rangle \approx \sum \beta_i \langle h^i_{RB}|h(\boldsymbol{\lambda})\rangle$$



Canizares et al PRD 2013

Reduced Order Quadratures

3) Fast likelihood computations

Online

- The cost of evaluating integrals scales lineally as the # of RBs m

$$\langle h(\lambda)|s\rangle_{ROQ} = 4\Re \sum_{k=1}^{m} \omega_k h(F_k; \vec{\lambda}) \quad \text{where} \quad w_j := \sum_{k=0}^{N} s^*(f_k) e_j(f_k) A^{-1} \Delta f$$

ROQ parameter estimation recipe

Reduced Order Quadratures

3) Fast likelihood computations

Online

•The cost of evaluating integrals scales lineally as the # of RBs m

$$\langle h(\lambda)|s\rangle_{ROQ} = 4\Re\sum_{k=1}^{m}\omega_k h(F_k;\vec{\lambda}) \quad \text{where} \quad w_j := \sum_{k=0}^{N} s^*(f_k)e_j(f_k)A^{-1}\Delta f$$

ROQ parameter estimation recipe

- Construct reduced basis: Find a set of templates that can reproduce every template in the model space to a certain specified precision. OFFLINE

Reduced Order Quadratures

3) Fast likelihood computations

Online

•The cost of evaluating integrals scales lineally as the # of RBs m

$$\langle h(\lambda)|s\rangle_{ROQ} = 4\Re\sum_{k=1}^{m}\omega_k h(F_k;\vec{\lambda}) \quad \text{where} \quad w_j := \sum_{k=0}^{N} s^*(f_k)e_j(f_k)A^{-1}\Delta f$$

ROQ parameter estimation recipe

- Construct reduced basis: Find a set of templates that can reproduce every template in the model space to a certain specified precision. OFFLINE
- Find empirical interpolation points: Find a set of points at which to match templates onto the basis. OFFLINE

Reduced Order Quadratures

3) Fast likelihood computations

Online

- The cost of evaluating integrals scales lineally as the # of RBs m

$$\langle h(\lambda)|s\rangle_{ROQ} = 4\Re\sum_{k=1}^{m}\omega_k h(F_k;\vec{\lambda}) \quad \text{where} \quad w_j := \sum_{k=0}^{N} s^*(f_k)e_j(f_k)A^{-1}\Delta f$$

ROQ parameter estimation recipe

- Construct reduced basis: Find a set of templates that can reproduce every template in the model space to a certain specified precision. OFFLINE
- Find empirical interpolation points: Find a set of points at which to match templates onto the basis. OFFLINE
- Construct signal specific weights: Compute the weights to use in the quadrature rule once data has been collected. startup

Reduced Order Quadratures

3) Fast likelihood computations

Online

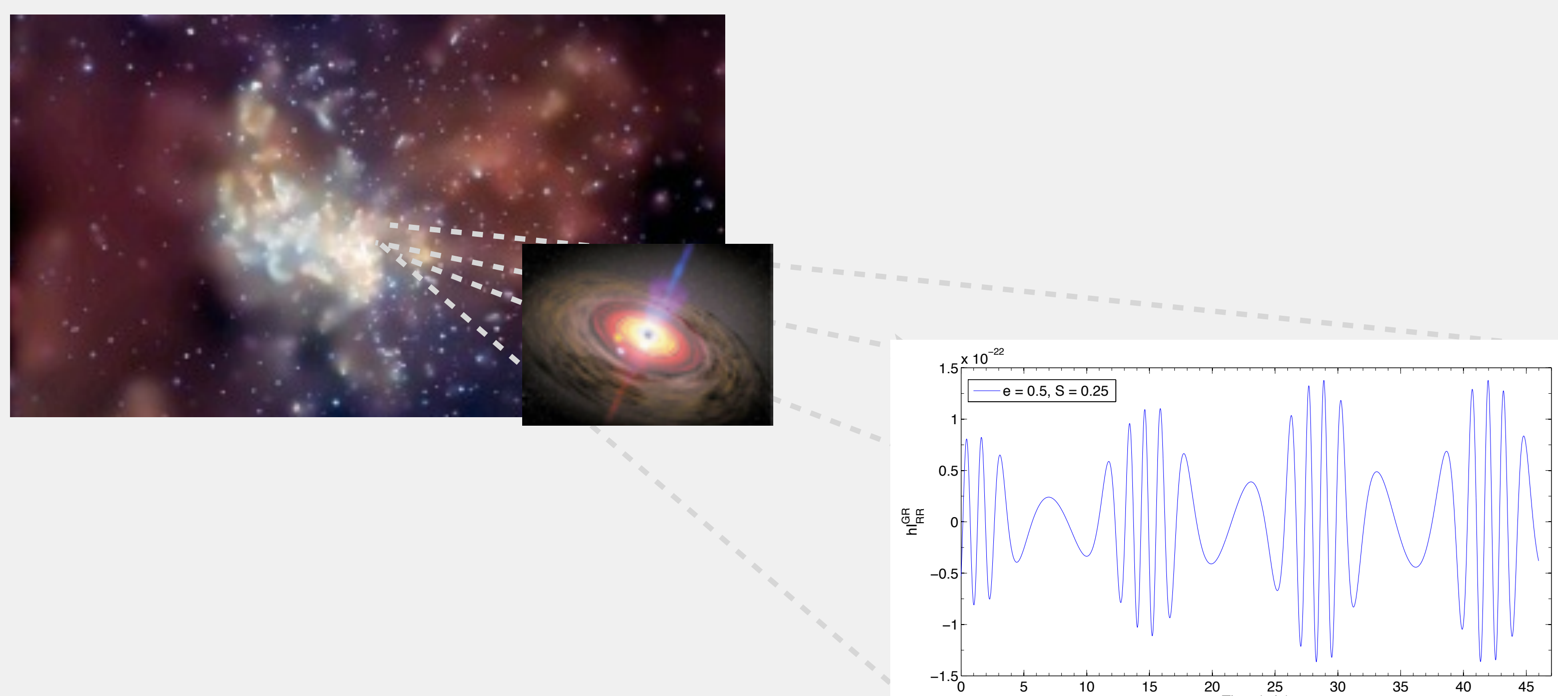
•The cost of evaluating integrals scales lineally as the # of RBs m

$$\langle h(\lambda)|s\rangle_{ROQ} = 4\Re\sum_{k=1}^{m}\omega_k h(F_k;\vec{\lambda}) \quad \text{where} \quad w_j := \sum_{k=0}^{N} s^*(f_k)e_j(f_k)A^{-1}\Delta f$$

ROQ parameter estimation recipe

- Construct reduced basis: Find a set of templates that can reproduce every template in the model space to a certain specified precision. OFFLINE
- Find empirical interpolation points: Find a set of points at which to match templates onto the basis. OFFLINE
- Construct signal specific weights: Compute the weights to use in the quadrature rule once data has been collected. startup
- Carry out parameter estimation: Evaluate likelihood/posterior over parameter space using ROQ rule and, e.g., MCMC. ONLINE

Parameter Estimation



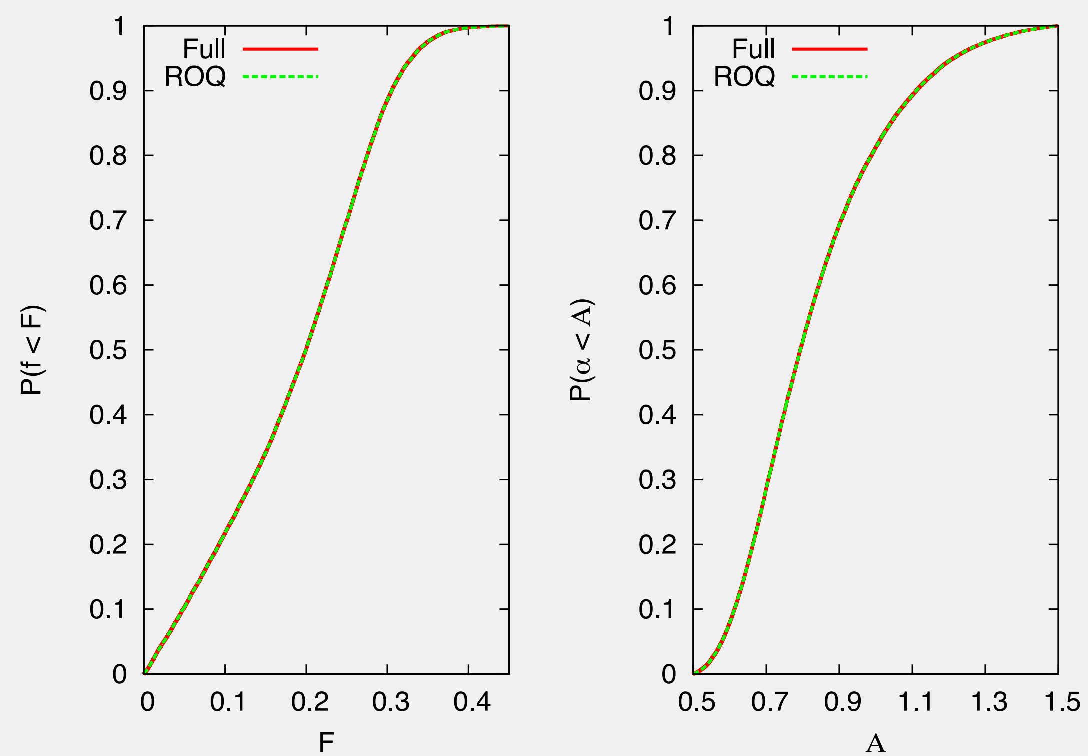
ROQ Parameter Estimation: Burst GW waveform

Two parameter case - burst width and central frequency.

		Recovered Values	
SNR	Method	f_0	α
5	Full	0.189 ± 0.095	0.831 ± 0.194
	ROQ	0.189 ± 0.095	0.831 ± 0.194
10	Full	0.172 ± 0.081	0.803 ± 0.136
	ROQ	0.172 ± 0.081	0.803 ± 0.136
20	Full	0.168 ± 0.075	0.800 ± 0.108
	ROQ	0.168 ± 0.075	0.800 ± 0.108
40	Full	0.212 ± 0.051	0.872 ± 0.091
	ROQ	0.212 ± 0.051	0.872 ± 0.091

Marginalised posterior distributions are indistinguishable and pass KS test with $p<10^{-6}$.

Factor of >20 speed up in MCMC run time.



Canizares et al PRD 2013

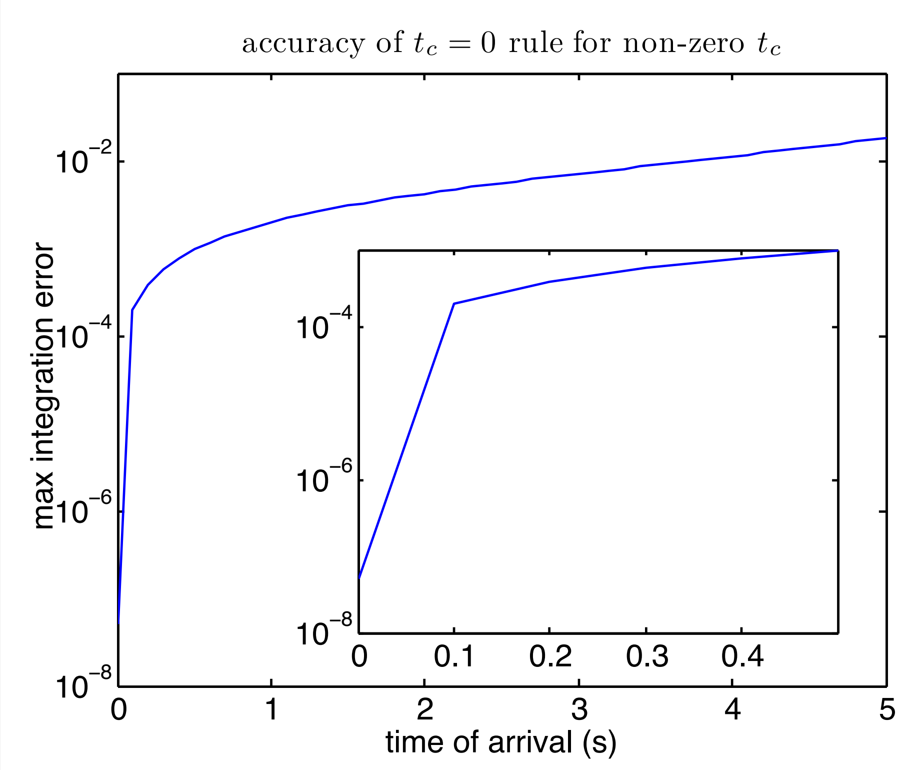
ROQ Parameter Estimation: Burst GW waveform

Full parameter space: 4D case A, α, f_0, t_c

- Amplitude which is multiplicative: Use same ROQ rule.
- Coalescence time: Introduces frequency-dependent phase shift. However, t_c=0 ROQ rule works well enough:

$$\langle s, h(\cdot; t_c)\rangle = \omega(t_c)^T h(\vec{F};0) = \omega(0)^T h(\vec{F};0) + t_c \frac{\partial \omega(t_c)^T}{\partial t_c}\Big|_{t_c=0} h(\vec{F};0) + O(t_c^2)$$

In the burst case the basis built for $t_c = 0$ works well for non-zero t_c.



Canizares et al PRD 2013

ROQ Parameter Estimation: Burst GW waveform

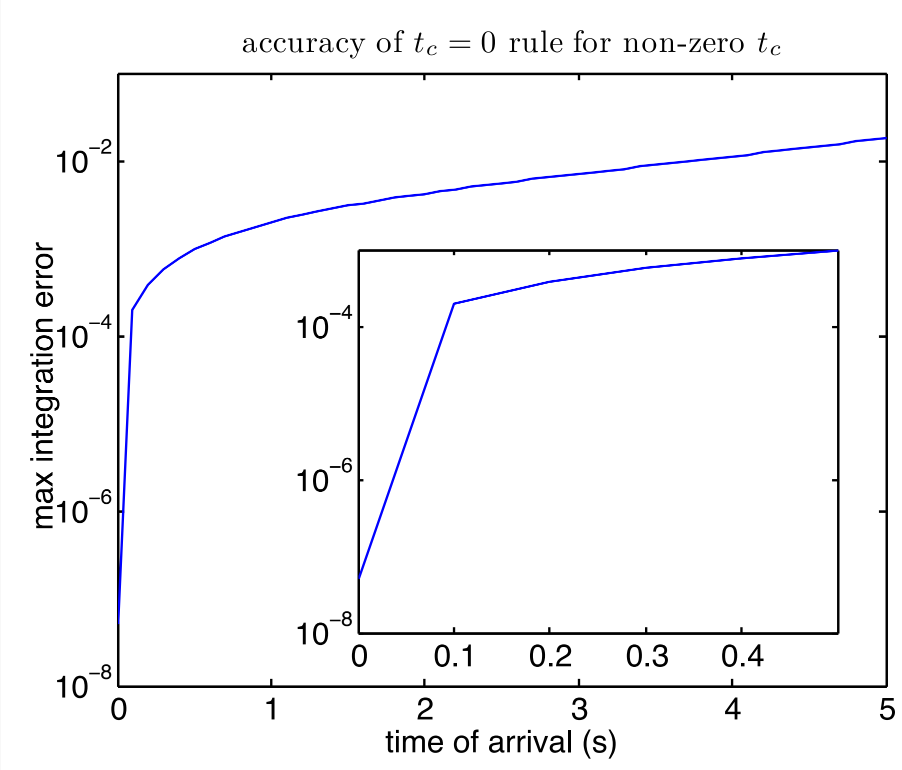
Full parameter space: 4D case A, α, f_0, t_c

- Amplitude which is multiplicative: Use same ROQ rule.
- Coalescence time: Introduces frequency-dependent phase shift. However, t_c=0 ROQ rule works well enough:

$$\langle s, h(\cdot; t_c)\rangle = \omega(t_c)^T h(\vec{F};0) = \omega(0)^T h(\vec{F};0) + t_c \frac{\partial \omega(t_c)^T}{\partial t_c}\Big|_{t_c=0} h(\vec{F};0) + O(t_c^2)$$

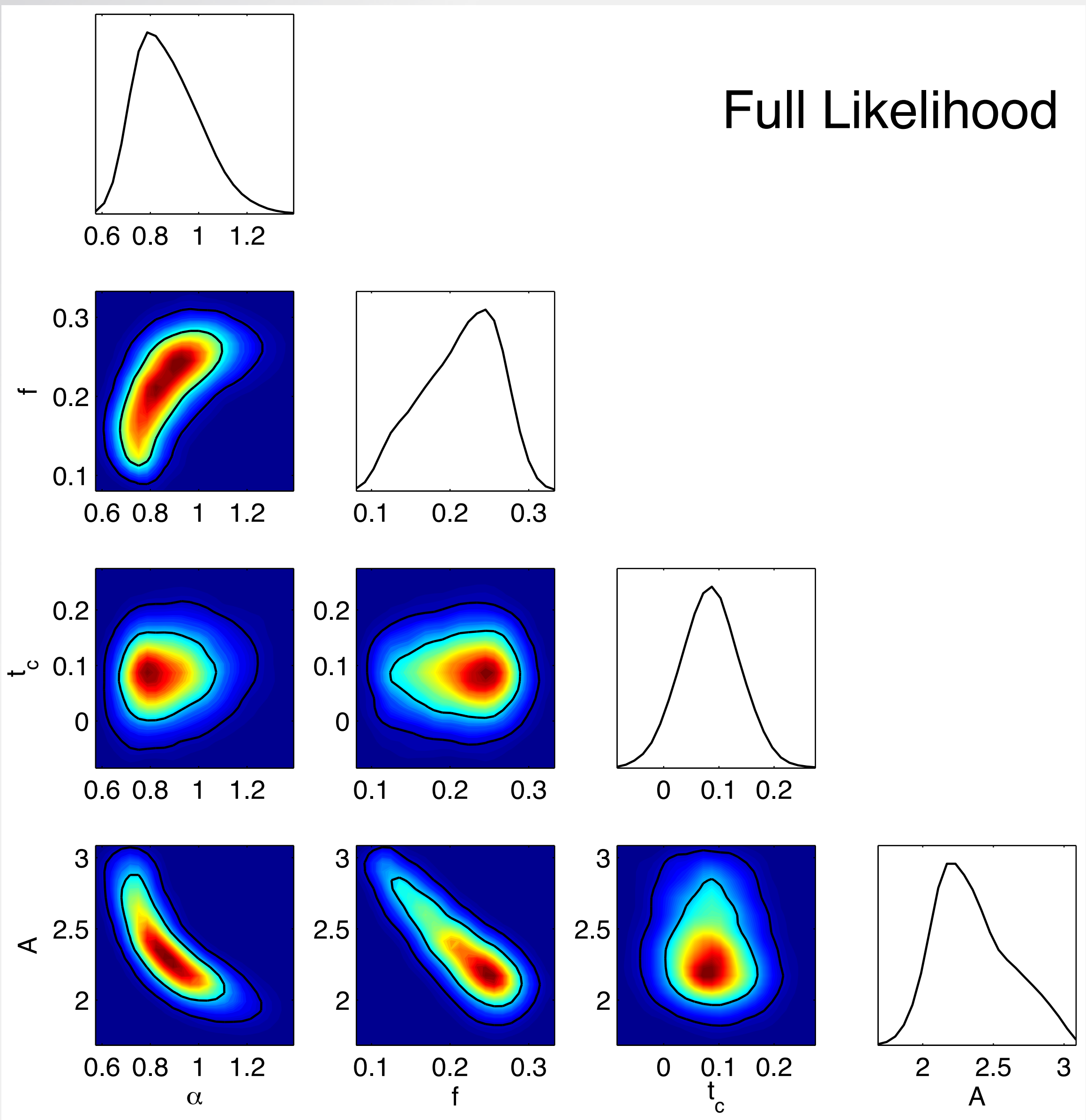
In the burst case the basis built for t_c = 0 works well for non-zero t_c.

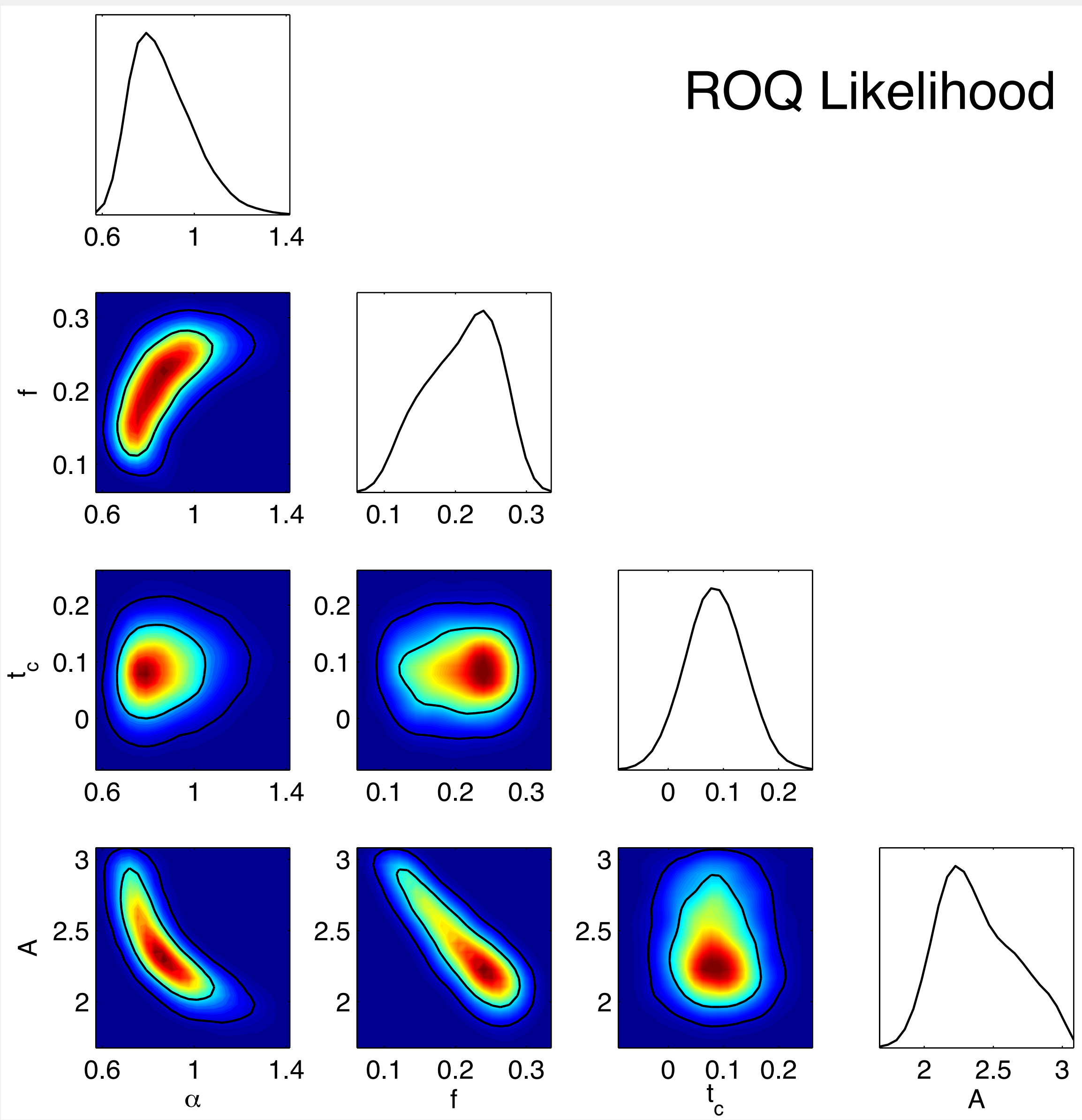
SNR	Method	f_0	α	t_c	A
		Recovered values			
5	Full	0.217 ± 0.069	0.896 ± 0.194	0.068 ± 0.104	1.704 ± 0.379
	ROQ	0.217 ± 0.068	0.897 ± 0.196	0.069 ± 0.104	1.702 ± 0.375
10	Full	0.212 ± 0.048	0.875 ± 0.132	0.084 ± 0.053	2.362 ± 0.278
	ROQ	0.209 ± 0.050	0.866 ± 0.132	0.085 ± 0.052	2.387 ± 0.287
20	Full	0.225 ± 0.029	0.891 ± 0.093	0.092 ± 0.028	2.944 ± 0.176
	ROQ	0.224 ± 0.029	0.892 ± 0.093	0.093 ± 0.028	2.944 ± 0.177
40	Full	0.248 ± 0.009	0.981 ± 0.041	0.097 ± 0.016	3.471 ± 0.157
	ROQ	0.248 ± 0.009	0.981 ± 0.042	0.097 ± 0.016	3.471 ± 0.157



Canizares et al PRD 2013

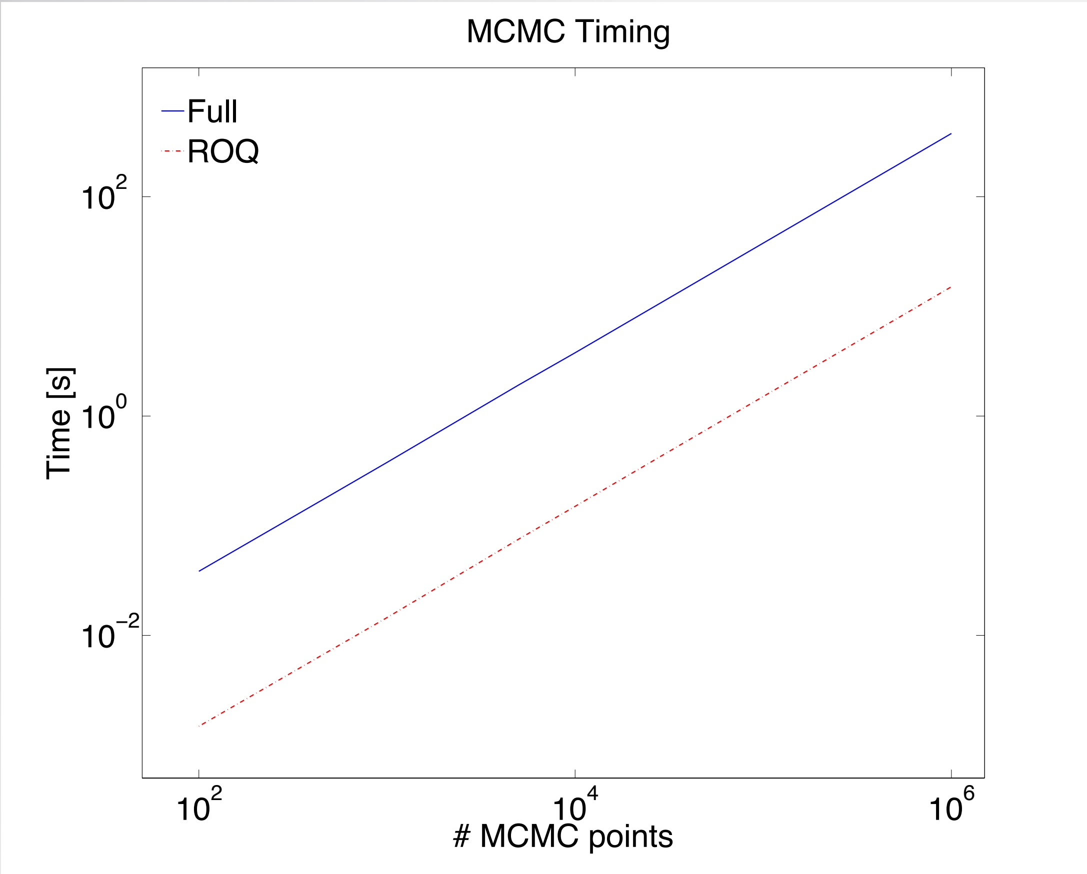
ROQ Parameter Estimation: Burst GW waveform

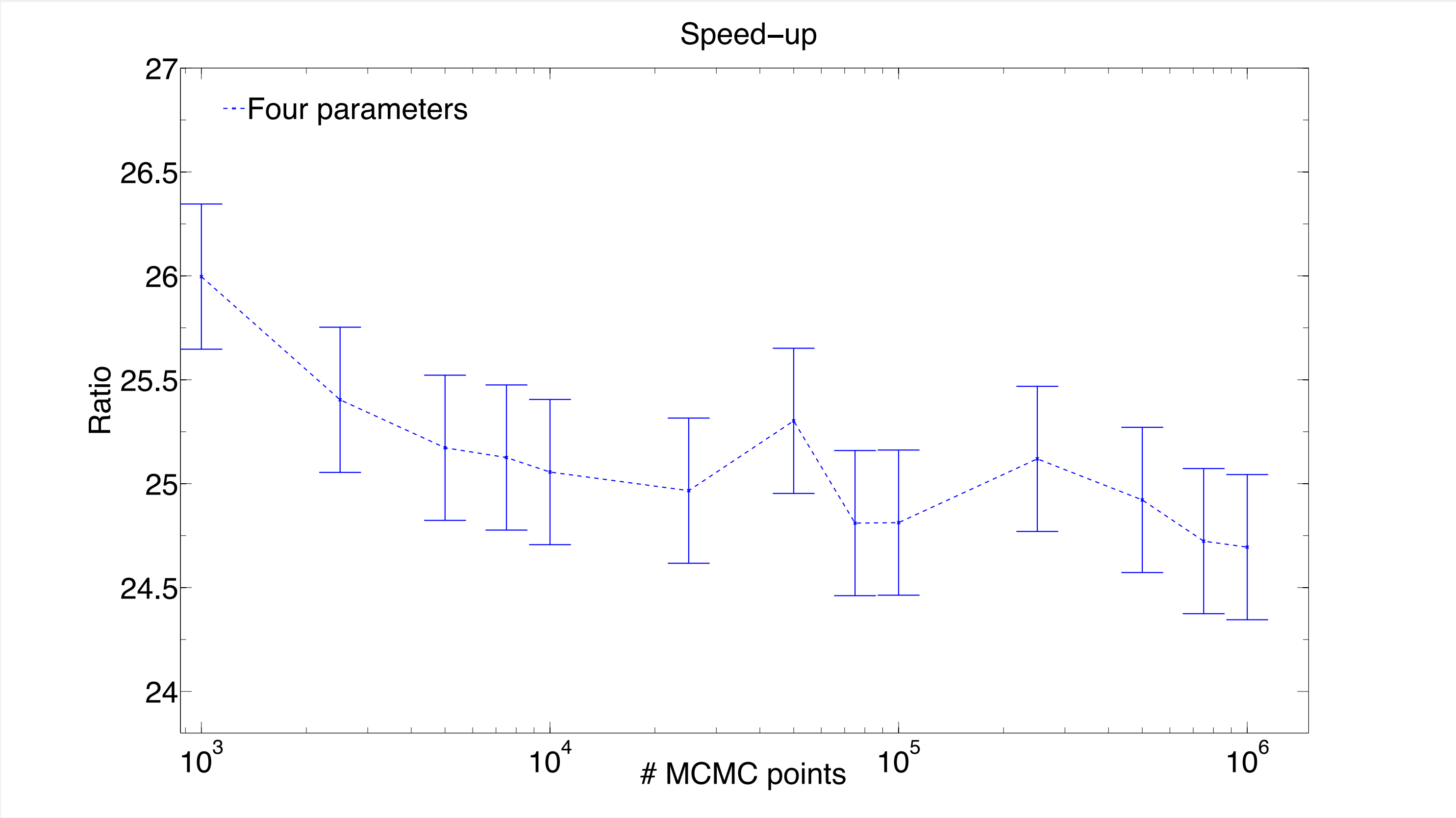




Canizares et al PRD 2013

ROQ Parameter Estimation: Burst GW waveform





Canizares et al PRD 2013

On going work: TaylorF2 waveforms

$$h(f,\lambda) = \mathcal{A} f^{-7/6} e^{i\psi_{3.5}^{(F2)}}$$

Waveform parameters $\lambda = \{m_1, m_2, t_c, \phi_c\}$

$$\Psi_{3.5}^{(F2)} = \Psi_{3.5}^{(F2)}(f, m_1, m_2, t_c, \phi_c)$$

$$\mathcal{A} = \mathcal{A}(m_1, m_2)$$

Standard MCMC simulation can take days to months depending on the observation times and sampling rates

On going work: TaylorF2 waveforms

$$h(f,\lambda) = \mathcal{A} f^{-7/6} e^{i\psi_{3.5}^{(F2)}}$$

$$\Psi_{3.5}^{(F2)} = \Psi_{3.5}^{(F2)}(f, m_1, m_2, t_c, \phi_c)$$

$$\mathcal{A} = \mathcal{A}(m_1, m_2)$$

Waveform parameters $\lambda = \{m_1, m_2, t_c, \phi_c\}$

Standard MCMC simulation can take days to months depending on the observation times and sampling rates

Expected speed-up, by comparing the ratio of N/m , is of 35. for early aLIGO (lower f= 40 Hz and observation time Tobs = 32s). If lower f = 10 Hz is achieved then Tobs = 30 minutes for BNS signals, and the speed-up will be 10^3**!**

On going work: TaylorF2 waveforms

$$h(f,\lambda) = \mathcal{A} f^{-7/6} e^{i\psi_{3.5}^{(F2)}}$$

Waveform parameters $\lambda = \{m_1, m_2, t_c, \phi_c\}$

$$\Psi_{3.5}^{(F2)} = \Psi_{3.5}^{(F2)}(f, m_1, m_2, t_c, \phi_c)$$

$$\mathcal{A} = \mathcal{A}(m_1, m_2)$$

Standard MCMC simulation can take days to months depending on the observation times and sampling rates

Expected speed-up, by comparing the ratio of N/m , is of 35. for early aLIGO (lower f= 40 Hz and observation time Tobs = 32s). If lower f = 10 Hz is achieved then Tobs = 30 minutes for BNS signals, and the speed-up will be 10^3**!**

Issue: How to handle tc

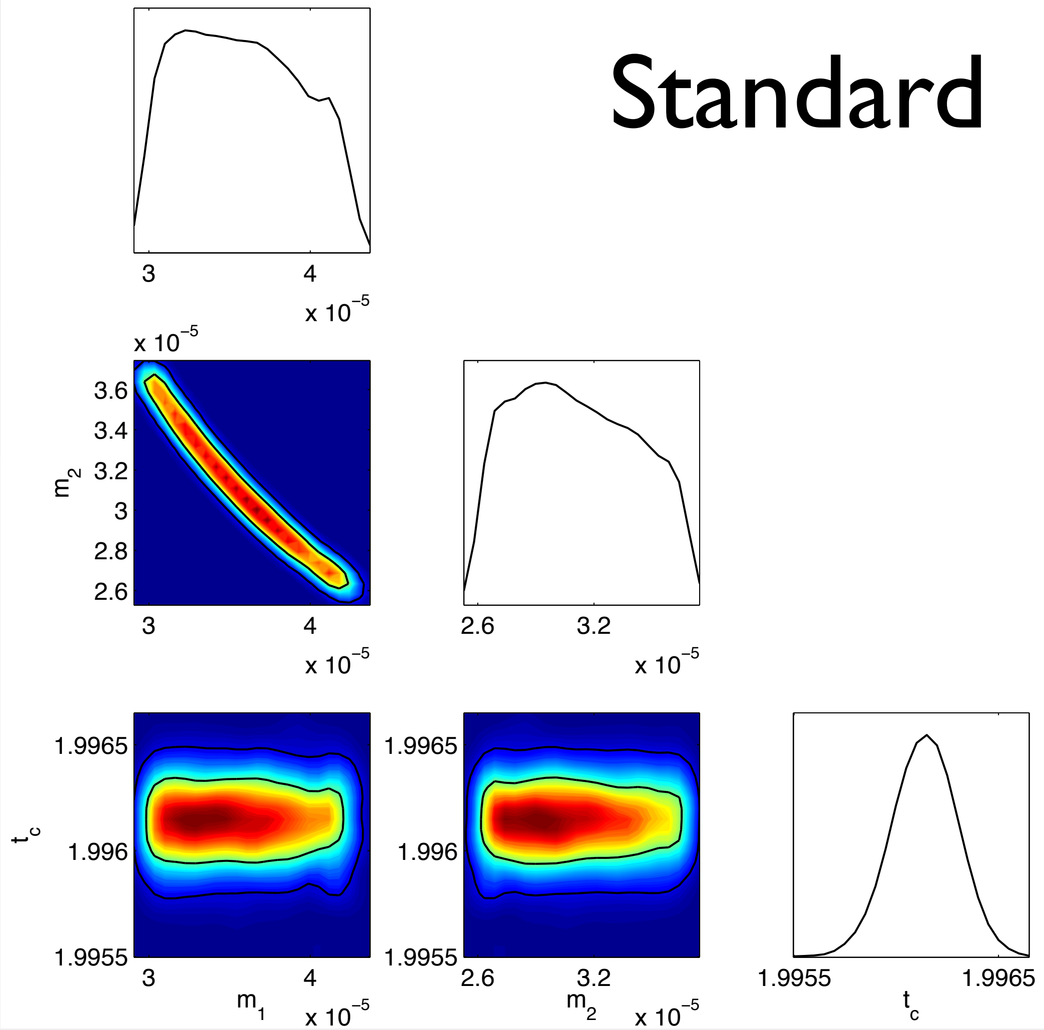
- Move the *tc* factor $e^{i2\pi f t_c}$ into the weight term, this avoids increase the number of RB due to a new (extrinsic) parameter.
- Use the DEIM interpolant based on the *tc* = 0 basis to (cheaply) evaluate the waveform at any frequency point - split the frequency dimension in several subdomains.

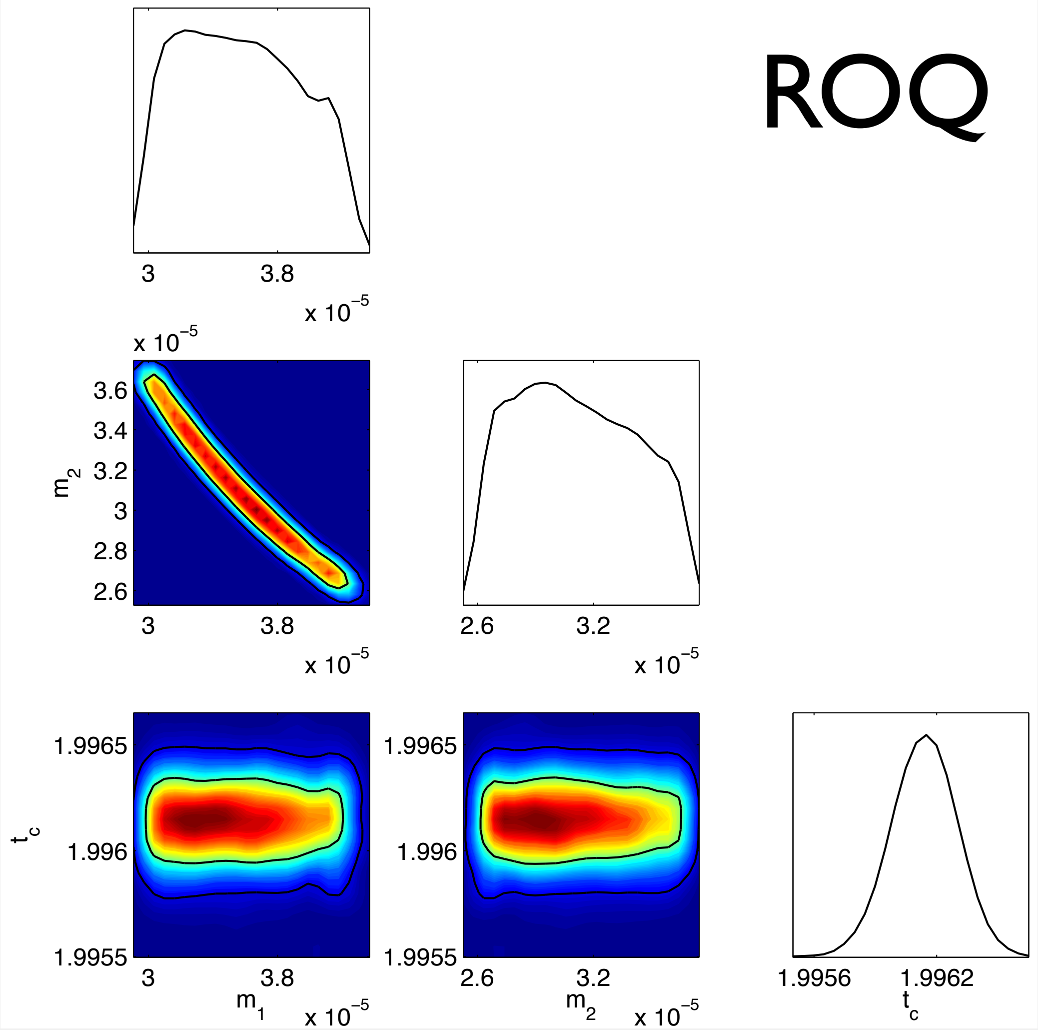
This approach gives different weights for each *tc* value leading to increased memory and offline.

Fresh first tests

On going work: TaylorF2 waveforms

First results at 0-PN





Further applications

- Sine-Gaussian searches are fast anyway - real advantage of ROQ is for expensive likelihoods.
- Method applies to any application in which overlaps/integrals need to be computed. Could also use it to speed-up searches over template banks, etc.
- Approach is not specific to gravitational wave applications - can be used for any data analysis problem in which likelihood overlap integrals need to be computed.

Summary

- ROQs speed-up likelihood evaluations for parameter estimation of a model

  1. Construct reduced basis for model space.
  2. Identify a set of empirical interpolation points at which templates are required to match basis.
  3. Construct a reduced order quadrature rule that reduces integral to a linear combination of template evaluations at the EIM points.

- In a gravitational wave data analysis context, have shown factors of ~20 speed-up for a toy model of sine-Gaussian waveforms.

- Now exploring application to compact binary coalescences and see speed-ups of a factor ~35 to 1,000.

diolch i chi!