Space Mapping Optimization and Model Reduction for Biogeochemical Models

A3: Algorithmic Optimal Control - C02 Uptake of the Ocean
(Prof. Dr. Thomas Slawig)

Malte Prieß  ○  mpr@informatik.uni-kiel.de
Cluster of Excellence “The Future Ocean”
General Research Aims

- The ocean is a key role in the climate system
- Already suffering from global warming
- Cover more than two thirds of our planet but yet little explored

The Aim:
Looking at past, present, future ocean changes
Investigating marine resources
Developing techniques for their sustainable use
Increase our understanding of ocean change and its potential and risks
Algorithmic Optimal Control - C02 Uptake of the Ocean

- **Natural causes + anthropogenic CO₂ emissions** \(\sim\) **global warming**

- **CO₂ concentration has doubled since 1900**

- To-date we assume 4 – 8°C in the business as usual case

- Agreement on the “2-degree-aim” until the year 2100

- This relates to a CO₂ emission reduction about 80% until 2050 (w.r.t. 1990)

- Concentrating only on a **sustainable energypolitics** will not comply with this aim

- Moreover we need to strongly think of **carbon management/sequestration** approaches
• The ocean is the biggest CO₂ sink
  More than half of anthropogenic CO₂ stored for long time
  Crucial impact on climate

• Natural Sequestration based upon global CO₂ cycle

• “Physical + Biological CO₂ pump” are the operators
  CO₂ can remain in the deep sea for years

• Ocean Circulation + Biogeochemical Models indispensable

Research Aims:
Reduce large uncertainties in existing biological models
  improve determination of current/future CO₂ sequestration potential
The Biogeochemical Models
Motivation

Present-day sea-surface nitrate concentrations (Conkright et al., 1994)

- Represent ecological processes contributing to global CO2 cycle
- Various models differing in complexity (# of state variables)
- Available data places significant limitations on complexity
- Nitrogen-based ecosystem model ↞ standard model
  ↞ 0-D transport-only to fully 3-D offline/online coupled physical-biological simulations
The Model Equations

- Linear transport/advection – diffusion eqs. with nonlinear forcing $q_i$

$$\frac{\partial y_i}{\partial t} = -v (\nabla y_i) + \nabla(\kappa \nabla y_i) + q_i(y, u, t)$$

- "Real world" simulation: coupling to ocean circulation models via the velocity field $v$ necessary (offline via TMM, online)

$y \in \{N, P, Z, D\} \in \mathbb{R}^{10^7}$

$u \in \mathbb{R}^{12}$

- $N = N(P, Z, D)$: dissolved inorganic nitrogen
- $P = P(N, Z)$: phytoplankton
- $Z = Z(P)$: zooplankton
- $D = D(P, Z)$: detritus
The Optimization (OPT) Problem
· Minimize distance between the model output $y(u_f)$ and the desired state $y_d$ (obs. data)

$$
\arg\min_{(y,u_f)} \mathcal{J}(y,u_f), \quad \mathcal{J}(y,u_f) = \left[ \frac{1}{2} \cdot \|y - y_d\|^2 + \frac{\alpha}{2} \cdot \|u_f - \bar{u}_f\|^2 \right]
$$

s.t. $e(y,u_f) = 0$ ; $u_l \leq u \leq u_u$

· Control variables are the unknown physical/biological parameters $u$ in the nonlinear coupling terms ($u$ stationary in time and space!)

```
+ Year 1991
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```

```
+ real data
+ model output
```
Space Mapping (SM) Optimization
Aims and First Definitions

- SM approach quit successfully applied for engineering models so far
- Seek at optimum of complex “fine” model
- SM drives the OPT of the fine model to a fast “coarse” model

~ Coarse model shares the same physics as the fine counterpart
~ avoiding computationally expensive fine model gradients and evaluations

- Key element is the mapping function (essential subproblem)
- Crucially depends on model similarity/ discrepancy
- Focus lies on the development of appropriate coarse models
fine model

\[ u_f^* = \arg\min_{u_f} H_f(u_f) := \frac{1}{2} \| f(t, u_f) - y \|^2 \quad : \quad \text{fine model optimum} \]

\[ u_f \in \Omega_f \subset \mathbb{R}^{n_f} \quad : \quad \text{control parameters} \]

\[ y_d \in \mathbb{R}^m \quad : \quad \text{desired state} \]

accurate but expensive, derivatives expensive/ not available

coarse model

\[ u_c^* = \arg\min_{u_c} H_c(u_c) := \frac{1}{2} \| c(t, u_c) - y \|^2 \quad : \quad \text{coarse model optimum} \]

\[ u_c \in \Omega_c \subset \mathbb{R}^{n_c} \quad : \quad \text{control parameters} \]

less accurate but fast, derivatives cheap
The SM Function

SM establishes mapping \( p : \Omega_f \rightarrow \Omega_c \) s.t.

\[
f(u_f) \simeq c[p(u_f)]
\]

\(p\) is defined as

misalignment function

\[
u_c = p(u_f) = \arg\min_{\hat{u}_c \in \Omega_c} r(\hat{u}_c, u_f), \quad r(u_c, u_f) = \frac{1}{2} \| c(u_c) - f(u_f) \|^2
\]

Now, replacing \( f \) by its surrogate \( c \circ p \), we obtain two SM approaches

\[
\tilde{u}_f^{(d)} = \arg\min_{u_f \in \Omega_f} \frac{1}{2} \| c[p(u_f)] - y \|^2 \quad \Leftrightarrow \quad p(\tilde{u}_f^{(p)}) - u_c^* = 0
\]

dual SM approach

primal SM approach

\( (*) \) Only satisfied under certain "ideal" conditions
Ideal Conditions

Let $\bar{U}_f$ and $\bar{U}_c$ be the sets of all SM solutions and coarse model minimizers

C1 $\bar{U}_c \subseteq p(\mathbb{R}^{n_c})$

C2 $\bar{U}_c \subseteq p(\bar{U}_f)$ (perfect mapping)

C3 $p$ is injective

C4 $\bar{U}_f$ and $\bar{U}_c$ are singletons

If conditions C1 - C4 hold we yield the ideal case:

$$\bar{u}_f^{(p)} = \bar{u}_f^{(d)} = u^* ; \quad p(u^*_f) = u^*_c$$
Example: The Aggressive SM (ASM) Algorithm

ASM just solves the **primal** SM problem

\[ \mathbf{F}(\mathbf{u}_f) = \mathbf{p}(\mathbf{u}_f) - \mathbf{u}_c^* \overset{!}{=} 0 \]  

(1)

by a quasi-Newton iteration and a Broyden rank-one update

\[
\begin{align*}
\mathbf{u}^{(k+1)}_f &= \mathbf{u}^{(k)}_f + \mathbf{s}^{(k)} , & \mathbf{B}^{(k)} \mathbf{s}^{(k)} &= -\mathbf{F}^{(k)} \\
\mathbf{B}^{(k+1)} &= \mathbf{B}^{(k)} + \frac{\mathbf{F}^{(k+1)}}{\mathbf{s}^{(k)}} \mathbf{s}^{(k)} \mathbf{s}^{(k)}^T , & \mathbf{F}^{(k)} := \mathbf{F}(\mathbf{u}^{(k)}_f) = \mathbf{p}(\mathbf{u}^{(k)}_f) - \mathbf{u}_c^*
\end{align*}
\]

Each step requires the evaluation of \( \mathbf{p} \), hence one fine model evaluation

\[
\mathbf{p}(\mathbf{u}^{(k)}_f) = \arg\min_{\mathbf{u}_c} \frac{1}{2} \| \mathbf{c}(\mathbf{u}_c) - \mathbf{f}(\mathbf{u}^{(k)}_f) \|^2
\]

More conveniently (1) is often using the least-square formulation

\[
\tilde{\mathbf{u}}_f = \arg\min_{\mathbf{u}_f} \| \mathbf{F}(\mathbf{u}_f) \|^2 ; \quad \mathbf{F}(\mathbf{u}^{k}_f + \mathbf{s}^{k}) \approx \mathbf{F}(\mathbf{u}^{k}_f) + \mathbf{B}^{(k)} \mathbf{s}^{k}
\]
The Coarse Models so Far
Coarse-Discretization Model

- Coarse-discretized (in time) version based upon the same model
- OPT of function $r(u_c, u_f)$ obtaining mapped parameter set, i.e.

$$p(u'_f) = u'_c = \arg\min_{u_c \in \Omega_c} r(u_c, u'_f)$$

- First approach: using simple steepest descent method

<table>
<thead>
<tr>
<th>factor n</th>
<th>$r(u'_f, u'_f)$</th>
<th>$r(u'_c, u'_f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>/</td>
</tr>
<tr>
<td>5</td>
<td>0.885</td>
<td>0.568</td>
</tr>
<tr>
<td>8</td>
<td>1.911</td>
<td>1.363</td>
</tr>
<tr>
<td>15</td>
<td>4.121</td>
<td>2.549</td>
</tr>
<tr>
<td>20</td>
<td>13.821</td>
<td>11.573</td>
</tr>
<tr>
<td>40</td>
<td>30.408</td>
<td>15.023</td>
</tr>
</tbody>
</table>

↷ Method seems to be unsuitable to obtain $p(u_f)$
↷ Switched over to MATLAB min. toolbox fmincon
Coarse-Discretization Model
Coarse-Discretization Model
Coarse-Discretization Model

\[ u^*_f = \arg\min_{u_f} H_f(u_f) \quad \leftarrow \quad \text{MATLAB fmincon + AD for } J_f \]

\[ u^*_c = \arg\min_{u_c} H_c(u_c) \quad \leftarrow \quad \text{MATLAB fmincon + AD for } J_c \]

\[ u_c = p(u_f) = \arg\min_{\hat{u}_c} r(\hat{u}_c, u_f) \quad \leftarrow \quad \text{MATLAB fmincon + AD for } J_c \]

Primal SM: \[ F(u_f) = p(u_f) - u^*_c \not\equiv 0 \quad \leftarrow \quad \text{Global Quasi-Newton SJN Method} \]

Jacobian of \( p \): \[ B^{(k)} \approx J_p(u_f^{(k)}) \quad \leftarrow \quad \text{Broyden rank-one approximation} \]
## Coarse-Discretization Model

<table>
<thead>
<tr>
<th>Iterate $u^{(k)}$</th>
<th>$u_f^{(0)} = u_c^{(0)}$</th>
<th>$H_f^{(0)}$</th>
<th>$H_f^{(0)}$</th>
<th>$H_f^{(0)}$</th>
<th>$H_f^{(0)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_f$</td>
<td>$H_f$</td>
<td>$1.58 \cdot 10^{-1}$</td>
<td>$1.58 \cdot 10^{-1}$</td>
<td>$1.58 \cdot 10^{-1}$</td>
<td>$1.58 \cdot 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$\delta u^{(k)}$</td>
<td>$2.14$</td>
<td>$2.14$</td>
<td>$9.22 \cdot 10^{-1}$</td>
<td>$9.22 \cdot 10^{-1}$</td>
</tr>
<tr>
<td>$u^*$</td>
<td>$H_f / H_f^{(0)}$</td>
<td>$-4$</td>
<td>$8.91 \cdot 10^{-2}$</td>
<td>$8.91 \cdot 10^{-2}$</td>
<td>$8.91 \cdot 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$\delta u^{(k)}$</td>
<td>$-4$</td>
<td>$9.22 \cdot 10^{-1}$</td>
<td>$9.22 \cdot 10^{-1}$</td>
<td>$9.22 \cdot 10^{-1}$</td>
</tr>
<tr>
<td>$\bar{u}_f$</td>
<td>$H_f / H_f^{(0)}$</td>
<td>$1.03 \cdot 10^{-3}$</td>
<td>$3.63 \cdot 10^{-3}$</td>
<td>$3.63 \cdot 10^{-3}$</td>
<td>$3.63 \cdot 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>$\delta u^{(k)}$</td>
<td>$3.65 \cdot 10^{-1}$</td>
<td>$6.98 \cdot 10^{-1}$</td>
<td>$6.98 \cdot 10^{-1}$</td>
<td>$6.98 \cdot 10^{-1}$</td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td>$3683$ s</td>
<td>$2722$ s</td>
<td>$2722$ s</td>
<td>$2722$ s</td>
</tr>
<tr>
<td># Iter.</td>
<td></td>
<td>$24$</td>
<td>$37 + 4$</td>
<td>$37 + 4$</td>
<td>$37 + 4$</td>
</tr>
</tbody>
</table>

![Graph showing phy values over time](image)
Consider the coarse model $c$ as the truncated Fourier series at parameters $u_c$ (= first fourier coefficients)

$$u_c^* = \arg\min_{u_c} H_c(u_c) = \text{FFT}^{(tr)}(y_d)$$

$$u_c = p(u_f) = \arg\min_{\hat{u}_c, u_f} r(\hat{u}_c, u_f) = \text{FFT}^{(tr)}[f(u_f)]$$

$$p : \mathbb{R}^{12} \mapsto \mathbb{C}^{12}, \quad u_f \in \mathbb{R}^{12}, \quad u_c \in \mathbb{C}^{12}$$
Fourier-type Model

OPT run using ASM. Solid/ dashed lines: fine and coarse model output (one year) $f(u_f^{(k)})$, $c(u_c^{(k)})$ in iteration $k$ (in the order blue, green, red, cyan, magenta), black: optimal solution $f(u_f^*), c(u_c^*)$. Here $\| F^{(5)} \| / \| F^{(0)} \| \approx 0.06$, $H_f^{(5)} / H_f^{(0)} \approx 0.04$. 

Malte Prieß - 24/03/2010 - Cluster of Excellence “The Future Ocean”
Globalized (Quasi-) Newton Method
Globalization strategy

Level function: \[ T(u_f | A) := \frac{1}{2} \left\| AF(u_f) \right\|^2 \]

Descent direction: Newton direction \[ J_F(u_f) s^{(N)} = -F(u_f) \]

( since \( < \nabla T, s^{(N)} >_{A=I} = -F^T J_F J_F^{-1} F = -\|F\|^2 < 0 \) )

Linesearch: Find parameter \( \sigma \) s.t. \( T(u_f + \sigma \cdot s | A) \leq t_k(A) \cdot T(u_f | A) \)

(i) **Local minima** of the level function \( T(u_f | A = I) \) where \( \nabla T(u_f) = J_F^T F \), \( F \neq 0 \) and \( J_F \) singular

.rx { margin-bottom: 4pt; } rx

\[ \hookrightarrow \] One can show that the choice \( A = J_F^{-1} \) leading to the natural level function is more convenient

(ii) **Case** \( J_F \) ill-conditioned: thus perturbed step \( s^{(N)} \) or \( s^{(QN)} \) might lead to non-descent dir. and breakdown of the algorithm

\[ \hookrightarrow \] Apply rank-strategy (yielding a descent direction for \( T(u_f | A = J_F^{-1}) \))
Original vs. Natural Level Function

Natural level function $h$ as function of two parameters $5, 1$ in optimum $x_f$.
Simple Testcases

Our focus mainly lies on two global methods:

(i) Global Quasi-Newton SJN method (cf. Kosmol, 1993)

Tests with a simple North Atlantic Boxmodel show:

Method (i)

\( \forall \) Local min. for \( \approx 40 \% \) of randomly choosen initial parameters
(similar “bad” results for other simple test functions)

Method (ii)

\( \forall \) Results follow

Method (ii) + rank-strategy

\( \forall \) Results follow
Open Problems
- Main focus should lie on the development of “appropriate” coarse models
  
  - Coarse-discretization model \(\sim\) Multigrid methods
  
  - Linearization of model equations. Where does the focus lie?
  
  - Furthermore what possible approaches could be done?

- Suitable validation techniques of the coarse models

- Error analysis

- Where exactly (w.r.t. the algorithms) is reduction of computer time consumption founded?
· Appropriate SM approaches and involved numerical/optimization methods

· Direct (primal through NLE) vs. indirect SM approach (dual through replacing $f$ by its surrogate $c[p(u_f)]$)?

· Adjoint approach for optimal control of a coarse model?

· Multipoint PE, implicit SM, other Jacobian approximation, regularization within $p$,

· TRASM, Hybrid SM methods

...
Thank you for your attention
[1] John W. B, Qingsha S. Cheng, Sameh A. Dakrouy, Ahmed S. Mohamed, Student Member, Student Member, Student Member, Mohamed H. Bakr, Kaj Madsen, and Jacob Søndergaard. Space mapping: The state of the art. 2004.


