An input/output model-reduction-based optimization method for the efficient design of large-scale engineering systems
Eduardo Luna-Ortiz and Constantinos Theodoropoulos

Motivation

Accurate process models are essential for:
- Process Design
- Optimisation
- Control

In many cases, however, models are available in a “packaged” black-box form

Model equations are unavailable in closed form or they even do not exist!

Can only run the code to obtain steady-state
Computationally intensive
Not always feasible (unstable steady states)

Furthermore
Optimisation, control can be impossible tasks for black-box codes

• Commercial packages, legacy codes, executables
  <Based on fixed-point procedures
  timesteppers, iterative solvers
• Microscopic, multi-scale simulators

Can use this low-dimensional subspace to perform ALL the “expensive” computations

Recursive Projection Method (RPM)

1. Identifies the dominant part of the system
   Typically low-dimensional

2. Apply Newton’s method on the low-dimensional part to accelerate convergence of the solver towards a steady state

3. It can be built around ANY existing black-box model, for which equations are not explicitly available

4. We can then perform gradient-based tasks: steady state computation, stability/bifurcation analysis

Our Objective

Develop a novel computational structure for the optimization of large-scale black-box process models and microscopic simulators using conventional computational resources.

Accurate process models are essential for:
- Process Design
- Optimisation
- Control

In many cases, however, models are available in a “packaged” black-box form

Model equations are unavailable in closed form or they even do not exist!

Can only run the code to obtain steady-state
Computationally intensive
Not always feasible (unstable steady states)

Furthermore
Optimisation, control can be impossible tasks for black-box codes

• Commercial packages, legacy codes, executables
  <Based on fixed-point procedures
  timesteppers, iterative solvers
• Microscopic, multi-scale simulators

Can use this low-dimensional subspace to perform ALL the “expensive” computations

Recursive Projection Method (RPM)

1. Identifies the dominant part of the system
   Typically low-dimensional

2. Apply Newton’s method on the low-dimensional part to accelerate convergence of the solver towards a steady state

3. It can be built around ANY existing black-box model, for which equations are not explicitly available

4. We can then perform gradient-based tasks: steady state computation, stability/bifurcation analysis

Our Objective

Develop a novel computational structure for the optimization of large-scale black-box process models and microscopic simulators using conventional computational resources.

we can also use the low-dimensional subspace to develop a gradient-based algorithm for the optimization of black-box models

BUT

Jacobians, Hessians even design equations are not available !!!
If the system has few decision variables (degrees of freedom) we can use Reduced Hessian methods (based on Sequential Quadratic Programming - SQP)

- Restricts the parameter search to the subspace of the degrees of freedom
- Computation of small Hessian matrices

\[ \text{Numerical computation is extremely demanding} \]

**Reduced Hessian**

\[ \text{Double reduction step:} \]
1. From \( n + \text{dof} \) to \( m + \text{dof} \)
2. From \( m + \text{dof} \) to \( \text{dof} \)

Compute only \( m \) reduced Hessians \( B \) (\( \text{dof} \times \text{dof} \))

**Small QP Subproblem**

Compute search direction \( \Delta z^j \) (\( \text{dof} \))
Project to \( m \) variables: \( \Delta p^j \)

Compute \( m \) Lagrange multipliers

Update solution for \( m \) variables
\[ p^j = p^j + \Delta p^j \]
Project to \( n \) variables
\[ x^k = x^k + \Delta x^k \]

**Model reduction technique:** calculate numerically only a few small Jacobians and Hessians by-passing all the expensive computations

新型方法：结合RPM和减少Hessian方法

**Case Study - Tubular Reactor**

Pseudohomogeneous model with axial dispersion:

\[ \frac{\partial x_1}{\partial t} = \frac{1}{Pe_1} \frac{\partial^2 x_1}{\partial z^2} - \frac{\partial x_1}{\partial z} \frac{\partial x_2}{\partial z} + Da(1-x_1) \exp \left( \frac{x_1}{1 + \frac{x_1}{y}} \right) \]

\[ \frac{\partial x_2}{\partial t} = \frac{1}{LePe_2} \frac{\partial^2 x_2}{\partial z^2} - \frac{1}{Le} \frac{\partial x_2}{\partial z} + B Da(1-x_1) \exp \left( \frac{x_2}{1 + \frac{x_2}{y}} \right) + \frac{Pc_2}{Le} \]

- \( x_1 \) dimensionless concentration
- \( x_2 \) dimensionless temperature
- \( Da \) Damkohler number
- \( Le \) Lewis number
- \( Pe_1 \) Peclet number for mass transport
- \( Pe_2 \) Peclet number for heat transport
- \( S \) dimensionless heat transfer coefficient
- \( B \) dimensionless adiabatic temperature rise
- \( x_{2w} \) dimensionless adiabatic wall temperature.
- \( z \) dimensionless length of reactor

**To test the method:**
- Dynamic model is solved using 4th order Runge-Kutta.
- Explicit method, Jacobians and Hessians are not explicitly available.
- This is equivalent to a black-box model.
Conclusions

An novel optimisation methodology has been developed aimed towards the optimisation of black-box large-scale process systems. Such systems are common in the chemical industries and include legacy codes, packaged software-based models and microscopic simulators. Reduced Hessian methods have been coupled with the Recursive Projection Method to construct an efficient gradient-based input/output optimization algorithm for these cases where Jacobians, Hessians and/or model equations are not available. Only small Jacobians and Hessians are computed numerically.

Future Work

- Exploit RPQP for optimization of large-scale commercial simulation software, such as CFD software
- Use RPQP in conjunction with iterative solvers
- Exploit RPQP for optimization of microscopic systems, i.e. Monte Carlo simulation of surface reactions
- Combine RPQP with branch and bound methods to enhance efficiency of MINLP solvers
- Explore methods of Global Optimization within the RPQP computational structure

Task: Maximize x2 at the exit by finding optimal Da number
Combine Runge-Kutta-based model with RPQP algorithm

Steady-States of x2 for a range of Da

Pe_i=5.0, Pe_o=5.0
Le=1.0, x_o=0.0
γ=20.0, B=12.0
β=1.5

RPQP performance for m=10 and m=8

Future Work

- Exploit RPQP for optimization of large-scale commercial simulation software, such as CFD software
- Use RPQP in conjunction with iterative solvers
- Exploit RPQP for optimization of microscopic systems, i.e. Monte Carlo simulation of surface reactions
- Combine RPQP with branch and bound methods to enhance efficiency of MINLP solvers
- Explore methods of Global Optimization within the RPQP computational structure

Task: Maximize x2 at the exit by finding optimal Da number
Combine Runge-Kutta-based model with RPQP algorithm

Comparison of performance for conventional SQP optimiser (using steady state model with analytical derivatives) and RPQP (m=10)

Case Study - Results

Task: Maximize x2 at the exit by finding optimal Da number
Combine Runge-Kutta-based model with RPQP algorithm

Steady-States of x2 for a range of Da

Pe_i=5.0, Pe_o=5.0
Le=1.0, x_o=0.0
γ=20.0, B=12.0
β=1.5

RPQP performance for m=10 and m=8

Comparison of performance for conventional SQP optimiser (using steady state model with analytical derivatives) and RPQP (m=10)

Optimal temperature profile

Pe_i=5.0, Pe_o=5.0
Le=1.0, Da=0.1
γ=20.0, B=12.0
β=1.5

Tubular reactor with three temperature actuators (Act)

Optimal solution

x1* = 0.9914
Act1* = 2.05
Act2* = 0.66
Act3* = 4.0

Optimal reactor temperature and jacket temperatures profiles

Optimal conversion profile