A NOVEL FILTER TRUST-REGION ALGORITHM FOR CONSTRAINED OPTIMIZATION USING REDUCED ORDER MODELING

Anshul Agarwal[†] and Lorenz T. Biegler[*]
[†]The Dow Chemical Company, Midland, MI
[*]Chemical Engineering Department, Carnegie Mellon University, Pittsburgh, PA

Abstract

Reduced order models (ROM) lead to powerful techniques to address computational challenges in PDE-constrained optimization. However, when incorporated within optimization strategies, ROMs are sufficiently accurate only in a restricted zone and thus, need to be systematically updated over the course of the optimization. As an enabling strategy, trust-region methods provide an excellent adaptive framework for ROM-based optimization. This study develops a novel filter trust-region algorithm for constrained optimization problems, which utilizes ROM refinement and a feasibility restoration phase. The algorithm not only restricts the optimization step within ROM's validity, but also synchronizes ROM updates with the information obtained during the course of optimization, thus providing a robust and globally convergent framework. When applied to the optimization of a two-bed four-step PSA system for CO_2 capture, it converges to a local optimum within reasonable CPU time.

Keywords

Reduced order modeling, Filter method, Trust-region

Introduction

Over the past decade, reduced order modeling (ROM) techniques based on proper orthogonal decomposition (POD) have been developed to generate cost-efficient representations of spatially and temporally distributed PDAEs (Kunisch and Volkwein, 1999, Armaou and Christofides, 2002). POD-based ROMs are formulated through the Galerkin projection of the PDAE system onto a truncated small set of POD basis functions, which lead to a significant reduction in the number of states as well as a much smaller optimization problem. However, for optimization, the ROM is accurate only at values of the decision variables where it is constructed ("root-point"), and the local nature of POD basis leads to inaccuracy of the ROM at other points in the decision variable space (Agarwal et al., 2009). Hence, the ROM needs to be updated as the optimization proceeds from the root-point to other points in the decision space.

Trust-region methods (Conn et al., 2000) offer an effective way to manage ROM updates over the course of optimization. These methods ensure that the step computed by the optimizer stays close to the root-point, and leads to ROM update decisions based on the information obtained during the optimization procedure. Trust-region methods for ROMs were first developed for unconstrained optimization. In particular, Alexandrov et al. (1998) developed an algorithm with a scaled objective function to ensure convergence to the correct optimum. Fahl (2000) developed the TRPOD algorithm based on an inexact gradient approach. Later, Alexandrov and coworkers (2001) extended their framework to incorporate equality and inequality constraints as well. In particular,

[*] To whom all correspondence should be addressed

they developed the MAESTRO-AMMO algorithm with an l_2 penalty function as a merit function.

In this work we develop a novel filter-based trust-region approach that extends the algorithm of Fletcher et al. (2002). We utilize a few concepts from Fahl's TRPOD algorithm and the MAESTRO-AMMO algorithm, and also incorporate Alexandrov's scaling scheme for the objective and constraints to ensure global convergence. In subsequent sections we present the details of the algorithm and apply it to optimize a two-bed four-step isothermal pressure swing adsorption (PSA) process to maximize CO_2 recovery from an N_2-CO_2 feed mixture.

Trust-region Subproblem with Constraints

We consider the nonlinear programming problem of the following form

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & c_E(x) = 0 \\ & c_I(x) \le 0 \\ & x^L \le x \le x^U \end{aligned} \tag{1}$$

where the objective function $f(x)$, equality constraints $c_E(x)$, and inequality constraints $c_I(x)$ are assumed to be sufficiently smooth and at least twice differentiable functions. The PDAE system is solved implicitly for given values of x. Solution profiles from the PDAEs are then used to compute the objective function and the constraints.

At iteration k of the optimization cycle, a ROM is constructed at a particular x_k, and this is used to build the model function for the trust-region subproblem. We define a ROM-based trust-region subproblem at iteration k as:

$$\begin{aligned} \min_s \quad & f_k^R(x_k + s) \\ \text{s.t.} \quad & c_{E,k}^R(x_k + s) = 0 \\ & c_{I,k}^R(x_k + s) \le 0 \\ & x^L \le x_k + s \le x^U \\ & \| s \|_\infty \le \Delta_k \end{aligned} \tag{2}$$

where $f_k^R(x_k + s)$ is the objective function and $c_{E,k}^R(x_k + s)$, and $c_{I,k}^R(x_k + s)$ are the equality and inequality constraints, respectively, computed from the reduced set of state variables of the ROM. We prefer to use a box type (l_∞) trust-region to restrict the step size.

To develop a robust and globally convergent trust-region algorithm involving ROMs, the following assumptions should hold (Conn et al., 2000)

(AF1) Functions $f(x)$, $c_E(x)$, and $c_I(x)$ are twice-continuously differentiable on $\Re^n$

(AF2) The function $f(x)$ is bounded below $\forall x \in \Re^n$

(AF3) The second derivatives of $f(x)$, $c_E(x)$, and $c_I(x)$ are uniformly bounded $\forall x \in \Re^n$

(A1) For iteration k, $f_k^R(x)$ is twice differentiable on B_k, where $B_k = \{x \in \Re^n \mid \| x - x_k \|_\infty \le \Delta_k\}, \quad \Delta_k > 0$

(A2) The value of the objective and the constraints for (1) and (2) coincide for every iterate k

$$f(x_k) = f_k^R(x_k) \qquad c_E(x_k) = c_{E,k}^R(x_k) \qquad c_I(x_k) = c_{I,k}^R(x_k)$$

(A3) The gradient of the objective and the Jacobian of the constraints for (1) and (2) coincide for every iterate k

$$\nabla f(x_k) = \nabla f_k^R(x_k) \qquad \nabla c_i(x_k) = \nabla c_{i,k}^R(x_k) \qquad i \in \{E, I\}$$

(A4) The second derivatives of $f_k^R(x)$, $c_{E,k}^R(x)$, and $c_{I,k}^R(x)$ remain bounded within B_k, for all k

Assumptions **(AF1)-(AF3)**, **(A1)**, and **(A4)** are assumed to hold. Assumption **(A2)** can be ensured by constructing accurate ROMs at each x_k. However, gradients of the objective and the constraints will in general differ for the ROM. To simplify the construction of ROMs that satisfy assumptions **(A2)** and **(A3)**, scaled (corrected) functions can be derived by using local corrections that correspond to the current iterate k. In this work, we define two additive correction schemes for the objective and constraints of (2):

Zero Order Correction (ZOC)

$$\tilde{\Phi}_k^R(x) = \Phi_k^R(x) + (\Phi(x_k) - \Phi_k^R(x_k)) \tag{3}$$

First Order Correction (FOC)

$$\begin{aligned} \tilde{\Phi}_k^R(x) = {} & \Phi_k^R(x) + (\Phi(x_k) - \Phi_k^R(x_k)) \\ & + (\nabla\Phi(x_k) - \nabla\Phi_k^R(x_k))^T (x - x_k) \end{aligned} \tag{4}$$

$$\Phi(x) = f(x), c_i(x) \qquad \Phi_k^R(x) = f_k^R(x), c_{i,k}^R(x) \qquad i \in \{E, I\}$$

Here ZOC satisfies only **(A2)** while FOC satisfies both **(A2)** and **(A3)**. Although ZOC does not satisfy **(A3)**, we still adapt it within our trust-region algorithm. We redefine (2) in terms of the corrected objective and constraints

$$\begin{aligned} \min_s \quad & \tilde{f}_k^R(x_k + s) \\ \text{s.t.} \quad & \tilde{c}_{E,k}^R(x_k + s) = 0 \\ & \tilde{c}_{I,k}^R(x_k + s) \le 0 \\ & x^L \le x_k + s \le x^U \\ & \| s \|_\infty \le \Delta_k \end{aligned} \tag{5}$$

While ZOC does not require derivative computation, and thus is cheap, FOC also requires $\nabla f(x_k)$, $\nabla c_E(x_k)$, and

$\nabla c_I(x_k)$ to be computed only once at x_k when the objective and the constraints are constructed for (5). The subproblem (5) is then solved using cheap derivatives of the ROM. However, these derivatives need to be computed every time whenever FOC is used to construct (5) for a trust-region iteration.

Filter-based Trust-region Algorithm

We develop a filter-based trust-region algorithm, which utilizes both ZOC and FOC for different parts of the algorithm. As constructing FOC for every trust-region iteration is expensive due to derivative computation, we apply ZOC when far from the optimum. Our algorithm thus begins with subproblems based on ZOC and later switches to FOC when no further improvement is observed. This work is patterned after Fletcher's trust-region filter method (Fletcher et al., 2002) with additional modifications for POD-based ROMs. Our proposed modifications also enjoy the global convergence properties of Fletcher's algorithm.

A filter method considers both minimization of the objective function $f(x)$ and constraint violation $\theta(x)$ as separate goals, where

$$\theta(x) = \max\left[\max_{i \in E}|c_i(x)|, \max_{i \in I} c_i(x)\right] \tag{6}$$

A *filter* is a list F of nondominated pairs (θ_i, f_i) such that for any two point (i, j) in filter F, either $\theta_i \le \theta_j$ or $f_i \le f_j$. During optimization, we move from x_k to $x_k + s_k$ only if the following condition holds:

$$\begin{aligned} &\theta(x_k + s_k) \le (1-\gamma_\theta)\theta_j \quad \text{or} \\ &f(x_k + s_k) \le f_j - \gamma_f \theta_j \end{aligned} \qquad \forall(\theta_j, f_j) \in F \cup (\theta_k, f_k) \tag{7}$$

where $\gamma_f, \gamma_\theta \in (0,1)$ are chosen to be small. We add (θ_k, f_k)–pairs to the filter for the acceptable iterates x_k.

Trust-region step computation

We decompose the trust-region subproblem (5) into a normal and a tangential subproblem. To minimize the constraint violation δ for ROM, we write the following normal subproblem

$$\tilde{\theta}^R(x) = \delta = \max\left[\max_{i \in E}|\tilde{c}_i^R(x)|, \max_{i \in I} \tilde{c}_i^R(x)\right] \tag{8}$$

$$\begin{aligned} \min_{v,\delta} \quad & \delta \\ \text{s.t.} \quad & -\delta \le \tilde{c}_{E,k}^R(x_k + v) \le \delta \\ & \tilde{c}_{I,k}^R(x_k + v) \le \delta \\ & x^L \le x_k + v \le x^U \\ & \|v\|_\infty \le \Delta_c, \qquad \delta \ge 0 \end{aligned} \tag{9}$$

For a non-zero tangential step, we choose $\Delta_c = 0.6\Delta_k$. With optimal infeasibility $\bar{\delta}$, we solve the following tangential subproblem to reduce the objective

$$\begin{aligned} \min_{s} \quad & \tilde{f}_k^R(x_k + s) \\ \text{s.t.} \quad & -\bar{\delta} \le \tilde{c}_{E,k}^R(x_k + s) \le \bar{\delta} \\ & \tilde{c}_{I,k}^R(x_k + s) \le \bar{\delta} \\ & x^L \le x_k + s \le x^U \\ & \|s\|_\infty \le \Delta_k \end{aligned} \tag{10}$$

We note that both subproblems are formulated with x_k as the center of the trust-region. This allows us to apply ZOC or FOC only once at x_k for both subproblems. In this work, we compute exact solution for both problems using the NPL solver IPOPT (Wächter and Biegler, 2006).

Switching condition

Relying solely on the condition (7) can cause sequence of iterates to provide sufficient reduction of $\theta(x)$ only, and not necessarily the objective. This could result in convergence to a feasible but suboptimal point. In order to prevent this, we use the following switching condition

$$\tilde{f}_k^R(x_k) - \tilde{f}_k^R(x_k + s_k) \ge \kappa_\theta \theta_k^{\gamma_s} \tag{11}$$

Here θ_k is the *actual* constraint violation from Eq. (6). If Eq. (11) fails, then the current θ_k is significant and we aim to improve that by inserting x_k to the filter. However, if Eq. (11) holds, then the reduction in the ROM-based objective function $\tilde{f}_k^R(x_k)$ is significant compared to current θ_k, and the algorithm should promote descent in the objective. In such a case, it is important that a sufficient decrease is also realized in the actual objective function $f(x)$. In other words, the following

$$\rho_k = \frac{ared_k}{pred_k} = \frac{f(x_k) - f(x_k + s_k)}{\tilde{f}_k^R(x_k) - \tilde{f}_k^R(x_k + s_k)} \ge \eta_1 \tag{12}$$

should hold together with Eq. (11). If this happens, we do not add x_k to the filter.

Eq. (11) ensures no feasible iterate is ever included in the filter. This is vital to not only avoid convergence to suboptimal points, but also for a finite termination of the feasibility restoration phase discussed later.

Algorithmic Parts ZOC and FOC

Since ROMs with ZOC are cheap to construct, our two-part algorithm begins in Part ZOC with problems (9) and (10) defined with ZOC, and proceeds until no further improvement in the objective or the infeasibility measure is obtained. After this, the algorithm moves to Part FOC where subproblems are constructed using first order corrections in Eq. (4).

Since constructing FOC is expensive, before switching to Part FOC we seek further improvement in the objective or infeasibility in Part ZOC by improving the accuracy of ROM. In particular, ROM is made more accurate by increasing the number of POD basis functions. POD subspace augmentation is carried out until an improved point is found or maximum limit for POD subspace dimension is reached.

Part FOC involves computing exact derivatives for each trust-region iteration. Because FOC ensures descent, we do not utilize ROM refinement. In fact, Part FOC allows working with smaller ROMs compared to Part ZOC as accurate steps can be generated. Also, once the algorithm proceeds from Part ZOC to Part FOC, it never returns to Part ZOC.

Feasibility restoration phase

The algorithm switches to a feasibility restoration phase if the new iterate either fails condition (7), or if it satisfies both Eq. (7) and Eq. (11) but doesn't provide sufficient decrease, i.e. $\rho_k < \eta_1$. The purpose of the restoration phase is to decrease the current constraint violation and generate a new iterate which is acceptable to the filter. In particular, it involves solving the normal subproblem repeatedly until such a point is obtained.

Whenever restoration is invoked at an iterate x_k, this point is added to the filter to avoid future visits. Restoration either generates a feasible iterate or converges to a local minimum of $\theta(x)$ indicating the problem might be infeasible.

Filter-based algorithm

Choose $0 < \eta_1 \le \eta_2 < 1 \le \eta_3, 0 < \gamma_1 \le \gamma_2 < 1 < \gamma_3, \kappa_\theta \in (0,1)$
$\gamma_f, \gamma_\theta \in (0,1), \beta \in (0,1), \gamma_s > 1/(1+\beta), \alpha_f, \alpha_\theta \in (0,1), \theta_{\max}$
$\Delta_0, \Delta_{\min}$, initial guess x_0, termination tolerance ε_0, Set $k = 0$

<u>Part ZOC</u>

1. Compute POD basis functions at x_k. Choose POD subspace size M_k and construct a ROM
2. *Step computation*
   - a. Solve subproblems (9) and (10) with ZOC
   - b. If $\theta_k \ge \theta_{\max}$, add x_k to filter and go to step 3
   - c. If $\Delta_k \le \Delta_{\min}$,
     - i. If $M_k \ge M_{\max}$ (ROM cannot be refined),
       - A. If $\theta_k = 0$, go to step 4
       - B. Add x_k to filter. Update M_k, go to step 3
     - ii. Refine ROM by increasing M_k, repeat step 2
   - d. If (7) fails, $x_{k+1} = x_k, \Delta_{k+1} = \gamma_1 \Delta_k$, increment k by 1 and repeat step 2
   - e. Compute ρ_k from Eq. (12)
   - f. If $pred_k < 0$ and $ared_k > 0$, go to 2(h)
   - g. If (11) holds and $\rho_k < \eta_1$, $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$, increment k by 1 and repeat step 2
   - h. If (11) fails, add x_k to the filter
   - i. Set $x_{k+1} = x_k + s_k$. If (11) fails, $\Delta_{k+1} = \Delta_k$, else

$$\Delta_{k+1} = \begin{cases} \gamma_2 \Delta_k & \text{if} \quad \rho_k \in [\eta_1, \eta_2), \\ \Delta_k & \text{if} \quad \rho_k \in [\eta_2, \eta_3), \\ \gamma_3 \Delta_k & \text{if} \quad \rho_k \ge \eta_3 \end{cases}$$

     Increment k by 1 and go to step 1
3. *Restoration with ZOC*
   - a. Solve normal problem (9) with ZOC until a point is found that satisfies (7). If found, add x_k to the filter, increment k by 1 and go to step 1, else continue
   - b. If $M_k \ge M_{\max}$ (ROM can't be refined), go to step 4
   - c. Refine ROM by increasing M_k, repeat step 3

<u>Part FOC</u>

4. Reinitialize Δ_k and M_k, construct ROM, go to step 6
5. Compute POD basis functions at x_k. Choose POD subspace size M_k and construct a ROM
6. *Step computation*
   - a. Solve subproblems (9) and (10) with FOC. If $\chi_k \le \varepsilon_0$ from (14), STOP
   - b. If $\theta_k = 0$ and $\Delta_k \le \Delta_{\min}$, STOP
   - c. If $\theta_k \ge \theta_{\max}$ or $\Delta_k \le \Delta_{\min}$ add x_k to filter, go to step 7
   - d. If (7) fails, $x_{k+1} = x_k, \Delta_{k+1} = \gamma_1 \Delta_k$, increment k by 1 and repeat step 6
   - e. Compute ρ_k from Eq. (12)
   - f. If (11) holds and $\rho_k < \eta_1$, $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$, increment k by 1 and repeat step 6
   - g. If (11) fails, add x_k to the filter
   - h. Set $x_{k+1} = x_k + s_k$. If (11) fails, $\Delta_{k+1} = \Delta_k$, else update Δ_k as in 2(i). Increment k by 1, go to step 5
7. *Restoration with FOC*: Solve normal problem (9) with FOC until (7) is satisfied. If such a point is

found, add x_k to filter, increment k and go to step 5, else STOP

The choice of the constants in the algorithm depends on the optimization problem and the scaling mechanism used for the decision variables. One peculiar feature of the algorithm is step 2(f) in Part ZOC. Even though $pred_k < 0$, this step allows us to move from x_k to $x_k + s_k$ because $ared_k > 0$. Such a scenario is possible especially with ROM-based trust-region subproblems without exact gradient information. Here $\rho_k < \eta_1$, so, if we move from 2(f) to 2(g), the step will be denied even though $ared_k > 0$. Hence, we jump from 2(f) to 2(h).

Another important feature of the algorithm is that in both sections, the trust-region radius is updated only when (11) holds. If (11) fails, the main effect of the current iteration is not to reduce the objective, but rather to reduce constraint violation (which is ensured by inserting x_k to the filter in steps 2(h) and 6(g)). In this case, we impose no further restriction on Δ_{k+1} and keep it the same as Δ_k.

For step 6(a), we define the first-order criticality measure χ_k in the following manner

$$\begin{aligned} \chi_k = \quad & \left| \min_d \quad \nabla \tilde{f}_k^R(x_k)^T d \right| \\ s.t. \quad & -\bar{\delta} \le \tilde{c}_{i,k}^R(x_k) + \nabla \tilde{c}_{i,k}^R(x_k)^T d \le \bar{\delta} \quad i \in E \\ & \tilde{c}_{i,k}^R(x_k) + \nabla \tilde{c}_{i,k}^R(x_k)^T d \le \bar{\delta} \quad i \in I \\ & \| d \| \le 1 \end{aligned} \tag{14}$$

We note that χ_k is defined in terms of ROM-based functions because of the first-order correction. Since the constraint set of (14) is convex and the objective function positive, it can be proved that χ_k is a first-order criticality measure, which vanishes only when x_k is a first-order critical point.

Part FOC is patterned after the SQP-filter algorithm proposed by Fletcher et al. (2002). All assumptions made by Fletcher et al. are satisfied and similar convergence properties hold. For a detailed convergence analysis, see Agarwal and Biegler (2011).

PSA Case Study

In this section, we apply the ROM-based filter trust-region algorithm to optimize a a two-bed four-step isothermal pressure swing adsorption process, as shown in Figure 1, with an 85%-15% N_2-CO_2 feed mixture. The operation consists of four distinct operating steps; pressurization, adsorption, depressurization (counter-current), and light reflux (or desorption). We maximize CO_2 recovery subject to a constraint on CO_2 purity. We consider five decision variables, high pressure P_h, low pressure P_l, step times t_p and t_a, and adsorption feed velocity u_a. The optimization problem is described below

$$\begin{aligned} \max \quad & CO_2 \text{ recovery} \\ \text{s.t.} \quad & CO_2 \text{ purity} \ge 50\% \\ & 1\,\text{bar} \le P_h \le 3\,\text{bar} \qquad 0.4\,\text{bar} \le P_l \le 1\,\text{bar} \\ & 35\,\text{sec} \le t_p \le 150\,\text{sec} \qquad 50\,\text{sec} \le t_a \le 400\,\text{sec} \\ & 0.1\,\text{m/s} \le u_a \le 0.3\,\text{m/s} \end{aligned} \tag{15}$$

A lower bound of 50% for CO_2 purity is reasonable as the cycle lacks any CO_2 enriching step and thus, we cannot achieve high purity with this cycle.

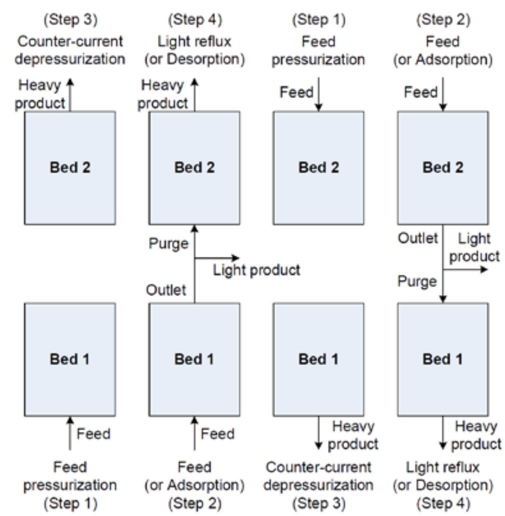

Figure 1: A two-bed four-step PSA cycle

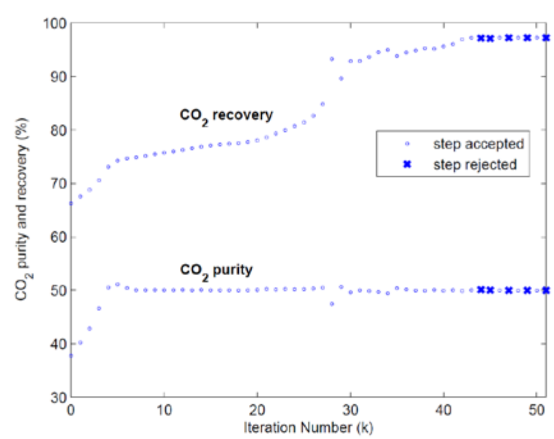

Figure 2: CO_2 purity and recovery

We develop a ROM for the PSA process and use it with the filter algorithm to solve problem (15). Figure 2 shows CO_2 purity and recovery with iterations, while Figure 3 shows iterations spent in Part ZOC and FOC, restoration phase iterations, and the progress of P_h and t_p. The algorithm begins in restoration phase in ZOC, which remains active till 4th iteration. Part ZOC then continues to improve the objective until 35th iteration. After that, ROM is not able to predict descent even after increasing POD subspace dimension. Hence, algorithm switches to Part FOC. The algorithm terminates after 51st iteration. We observe that the algorithm allows moves which increase infeasibility while improving objective. Such flexibility leads to convergence in fewer iterations.

A key observation is the value of t_p which increases steadily in Part ZOC but starts decreasing and hits the lower bound once Part FOC starts. We infer that the (ROM) gradient of the objective function with respect to t_p has an opposite sign during optimization in Part ZOC, which gets corrected in Part FOC. Once P_h reaches its upper bound, no further improvement is possible in the objective and thus Part ZOC terminates after 35th iteration due to this incorrect gradient.

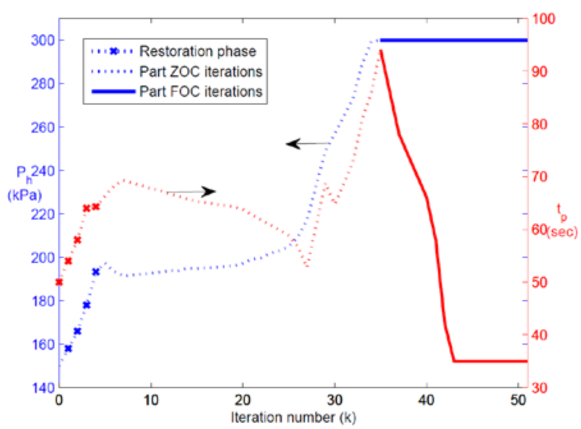

Figure 3: Iterations for Part ZOC and FOC

Table 1 lists the optimization results. Using finite difference gradients we conclude that the algorithm converged to a local optimum.

Conclusions

Trust-region based methodology provides an excellent adaptive framework to systematically utilize reduced-order models for optimization since it not only restricts the validity zone of the reduced-order model, but also provides a robust and globally convergent algorithm. In particular, we develop a filter-based trust-region framework since it allows steps that can achieve greater reduction in the objective by increasing infeasibility in a controlled manner. We follow a hybrid strategy and incorporate Part ZOC to avoid expensive gradient calculations. For the PSA case study, we observe that 35 iterations out of the total 51 are carried out in Part ZOC of the algorithm, which is quite encouraging as it delays expensive gradient evaluations for FOC. Thus, we infer that a hybrid strategy and POD subspace augmentation are potentially useful tools for optimization with ROMs. Future work will explore the improvement of this strategy through the addition of second order corrections and better tuning parameters.

Finally, the performance of the algorithm relies heavily on the quality of ROM and its ability to accurately predict the descent direction. In future, alternate methodologies will be explored to build efficient ROMs.

Table 1: Optimization results for PSA case study

No. of variables in ROM	52247
Trust-region iterations	51
Total CPU time	1.36 hrs
Optimal P_h	300 kPa
Optimal P_l	40 kPa
Optimal t_p	35 sec
Optimal t_a	187.91 sec
Optimal u_a	12.77 cm/sec
Optimal CO_2 purity	50.01%
Optimal CO_2 recovery	97.26%

References

Agarwal, A., Biegler, L. T., Zitney, S. E. (2009). Simulation and Optimization of Pressure Swing Adsorption Systems using Reduced-Order Modeling. *Ind. Eng. Chem. Res., 48,* 2327.

Agarwal, A., Biegler, L. T. (2011). A Trust-region Framework for Constrained Optimization using Reduced Order Modeling. *Optim. Eng.,* accepted for publication..

Alexandrov, N. M., Dennis Jr., J. E., Lewis, R. M., Torczon, V. (1998). A Trust-region Framework for Managing the Use of Approximate Models in Optimization. *Struct. Optim., 15,* 16.

Alexandrov, N. M., Lewis, R. M., Gumbert, C. R., Green, L. L., Newman, P. A. (2001). Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models. *J. Aircraft, 38,* 1093.

Armaou, A., Christofides, P. D. (2002). Dynamic Optimization of Dissipative PDE Systems using Nonlinear Order Reduction. *Chem. Eng. Sci., 7,* 5083.

Conn, A. R., Gould, N. M., Toint, P. L. (2000). Trust-region Methods. *MPS-SIAM.* Philadelphia, PA.

Fahl, M. (2000). Trust-region Methods for Flow Control based on Reduced Order Modelling. *PhD Thesis.,* Trier Univ.

Fletcher, R., Gould, N. M., Leyffer, S., Toint, P. L., Wächter, A. (2002). Global Convergence of a Trust-region SQP-Filter Algorithm for General Nonlinear Programming. *SIAM J. Optim., 13,* 635.

Kunisch, K., Volkwein, S. (1999). Control of Burgers Equation by a Reduced-Order Approach using POD. *J. Opt. Theory Appl., 102,* 345.

Wächter, A., Biegler, L. T. (2006). On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-scale Nonlinear Programming. *Math. Program., 106,* 25.