

Model Reduction-Based Constrained Optimization for Large-Scale Steady State Systems Using Black-Box Simulators

*Ioannis Bonis and Constantinos Theodoropoulos**

School of Chemical Engineering and Analytical Science, The University of Manchester, Manchester M60 QD, United Kingdom

Abstract

A novel model reduction-based deterministic optimization algorithm for large scale systems modeled by black-box simulators is presented. The proposed algorithm relies solely on the computation of low-dimensional Jacobian and reduced Hessian matrices, which correspond to the dominant modes of the system at hand. A basis for the dominant subspace of the system is computed using subspace iterations and is exploited for the calculation of the reduced Jacobian matrices through a small number of numerical directional perturbations. The reduced Hessian matrices are calculated from a 2-step projection scheme, firstly onto the dominant subspace of the system and secondly onto the subspace of the decision variables. To illustrate the efficiency of the proposed algorithm, we have applied it to the cases of the optimization of a tubular reactor, as well as a Counterflow Jet Reactor which is modeled using MPSalsa.

Introduction

The design of all industrial systems involves the concept of optimization. Many, if not most, of those systems are complex and are typically described accurately by a set of partial differential equations (PDEs). The latter are discretized over a mesh for the numerical simulation of the system at hand, which leads to a large-scale system. Steady state simulators employ iterative methods for the solution of those systems. Optimization of those processes can be based on deterministic [1], or stochastic/meta-heuristic [2] methods. Stochastic methods perform a large number of function evaluations, invoking the system simulator. Thus they are more appropriate for moderate-sized problems [2]. On the other hand, the application of deterministic optimization methods to large-scale systems with constraints is often problematic or even unrealistic, having increased requirements regarding computing power and memory size. The task is deemed even more tedious in the case of black-box solvers (commercial simulators or legacy codes), since the system equations are not available to the user.

In recent years we have developed a model reduction-based framework for gradient-based steady-state [3] and dynamic [4] optimization that employs input/output dynamic simulators. Here we extend this work by presenting a novel framework for steady-state optimization which uses black-box steady-state simulators based on solvers employing iterative linear algebra methods. We demonstrate the behavior of the algorithm by applying it for the optimization of a tubular reactor [5] and its performance and efficiency in handling large-scale input/output simulators through another illustrative example: the optimization of a Counter-Flow Jet Reactor [6] which is simulated using the state-of-the-art massively parallel finite element code MPSALSA developed at SANDIA National Laboratories [7].

* Corresponding author, e-mail: k.theodoropoulos@manchester.ac.uk

The proposed algorithm

The optimization problem considered here is:

$$\min f(\mathbf{x}) \quad \text{s.t. } \mathbf{G}(\mathbf{x}) = 0, \quad \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U \quad (1)$$

where $\mathfrak{R}^{N+dof} \rightarrow \mathfrak{R}$ is the objective function and $\mathbf{x} \in \mathfrak{R}^{N+dof}$ is the joint vector of the dependent $\mathbf{u} \in \mathfrak{R}^N$ and independent variables $\mathbf{z} \in \mathfrak{R}^{dof}$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{u} \\ \mathbf{z} \end{bmatrix} \quad (2)$$

Feasible states are calculated by solving the equation:

$$\mathbf{G}(\mathbf{x}) = 0 \quad (3)$$

using the input/output, black-box simulator.

If the dimension of \mathbf{x} is small to medium, the optimization problem can be efficiently handled by the Sequential Quadratic Programming method (SQP). The SQP is equivalent to solving the Kuhn-Karush-Tucker optimality conditions using the Newton-Raphson method. In every step of this method, a QP subproblem is solved:

$$\min_{\mathbf{d}} \frac{1}{2} \mathbf{d}^T \mathbf{B} \mathbf{d} + \nabla f(\mathbf{x})^T \mathbf{d}, \quad \text{s.t. } \nabla \mathbf{G}(\mathbf{x})^T \mathbf{d} = -\mathbf{G}(\mathbf{x}), \quad (\mathbf{x}^L - \mathbf{x}) \leq \mathbf{d} \leq (\mathbf{x}^U - \mathbf{x}) \quad (4)$$

Where $\mathbf{B} \in \mathfrak{R}^{n+dof \times n+dof}$ is the Hessian of the system and \mathbf{d} is the search direction ($\mathbf{d} \in \mathfrak{R}^{n+dof}$).

For large scale systems that include a small number of degrees of freedom, the reduced Hessian methods are more appropriate [8]. They consider decomposing the space of the search direction in two subspaces, one being the tangent space of the constraints $\nabla \mathbf{G}(\mathbf{x})$ and the other being its complement. Let \mathbf{Z} and \mathbf{Y} non-orthonormal bases for the two subspaces correspondingly:

$$\mathbf{Z} = \begin{bmatrix} -(\nabla_u \mathbf{G}^T)^{-1} \nabla_z \mathbf{G}^T \\ \mathbf{I} \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \quad (5)$$

This way, the search direction can be decomposed as follows:

$$\mathbf{d} = \mathbf{Y} \mathbf{p}_y + \mathbf{Z} \mathbf{p}_z \quad (6)$$

Following this formulation, the QP subproblem defined in Eq. 4 can be transformed as follows:

$$\min_{\mathbf{p}_z} \left(\mathbf{Z}^T \nabla f + \mathbf{Z}^T \mathbf{B} \mathbf{Y} \mathbf{p}_z \right)^T \mathbf{p}_z + \frac{1}{2} \mathbf{p}_z^T \left(\mathbf{Z}^T \mathbf{B} \mathbf{Z} \right) \mathbf{p}_z \quad \text{s.t. } (\mathbf{x}^L - \mathbf{x}) \leq \mathbf{Y} \mathbf{p}_y + \mathbf{Z} \mathbf{p}_z \leq (\mathbf{x}^U - \mathbf{x}) \quad (7)$$

The matrix $\mathbf{Z}^T \mathbf{B} \mathbf{Z} = \mathbf{B}_R$, $\mathbf{B}_R \in \mathfrak{R}^{dof \times dof}$ is the reduced Hessian and can be calculated numerically, generally through a BFGS update procedure. The Lagrange multipliers needed for this calculation can be calculated from:

$$\mathbf{Y}^T \mathbf{B} \mathbf{Y} \mathbf{p}_y + \mathbf{Y}^T \mathbf{B} \mathbf{Y} \mathbf{p}_z + \mathbf{Y}^T \nabla \mathbf{G} \lambda = -\mathbf{Y}^T \nabla f \quad (8)$$

Note here that the term \mathbf{p}_y is zero if feasible points (steady-states) are calculated at each iteration [8].

The calculation and inversion of the Jacobian for the computation of the basis Z as defined by Eq. 5 is expensive for large problems. Especially for the case of black-box simulators, the Jacobian may not be available to the user, or even not calculated if the solver employs linear algebra procedures. To tackle this problem, we consider using a reduced Jacobian rather than the full one.

Let \mathbf{P} be the invariant subspace of the Jacobian belonging to the m right-most eigenvalues. Those are the dominant eigenmodes of the system at hand which govern its behavior. In many engineering problems $m \ll n$. This subspace can be efficiently identified through its basis $\hat{Z} \in \mathfrak{R}^{n \times m}$, using subspace iterations. Here we used the subroutine EB22, part of the HSL library [9]. The Jacobian itself is not explicitly provided. Only products of the Jacobian with given vectors are needed, which are calculated using numerical directional perturbations on the constrains. The reduced Jacobian, $\mathbf{H} \in \mathfrak{R}^{m \times m}$, is a projection of the original one onto \mathbf{P} :

$$\mathbf{H} = \hat{Z}^T \mathbf{G}_u \hat{Z} \quad (9)$$

The dominant subspace of the system can be easily extended to include the independent variables. The basis for the new subspace is $\hat{Z}_{ext} \in \mathfrak{R}^{(N+dof) \times (m+dof)}$:

$$\hat{Z}_{ext} = \begin{bmatrix} \hat{Z} & 0 \\ 0 & I \end{bmatrix} \quad (10)$$

An approximation of a basis for the subspace of the independent variables, Z_r , can be constructed using the reduced Jacobian, \mathbf{H} :

$$Z_r = \begin{bmatrix} -\mathbf{H}^{-1} \hat{Z}^T \nabla_z \mathbf{G}^T \\ I \end{bmatrix} \quad (11)$$

Hence, a basis for the subspace that stems from the 2-step projection scheme is:

$$Z^* = \hat{Z}_{ext} Z_r = \begin{bmatrix} \hat{Z} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} -\mathbf{H}^{-1} \hat{Z}^T \nabla_z \mathbf{G}^T \\ I \end{bmatrix} = \begin{bmatrix} -\hat{Z} \mathbf{H}^{-1} \hat{Z}^T \nabla_z \mathbf{G}^T \\ I \end{bmatrix} \quad (12)$$

Using this base, the reduced Hessian of the Lagrangian, $B \in \mathfrak{R}^{dof \times dof}$, can be calculated with a few numerical directional perturbations:

$$\hat{\mathbf{B}}_R = Z^{*T} \mathbf{B} Z^* = Z_r^T \left(\hat{Z}_{ext}^T \mathbf{B} \hat{Z}_{ext} \right) Z_r \quad (13)$$

The Lagrange multipliers, ϕ , needed for this calculation are reduced and are projections of the original ones onto the dominant subspace: $\phi = Z^T \lambda$. Those are calculated from Eq. 14:

$$\mathbf{H} \phi = -\hat{Z}^T \mathbf{Y}^T \nabla f \quad (14)$$

It is straightforward to show that the QP subproblem of Eq. 7 becomes:

$$\min_{\mathbf{p}_z} \left(Z^T \nabla f_z \right)^T \mathbf{p}_z + 1/2 \mathbf{p}_z^T B_R \mathbf{p}_z, \text{ s.t. } (\mathbf{x}^L - \mathbf{x}) \leq Z \mathbf{p}_z \leq (\mathbf{x}^U - \mathbf{x}) \quad (15)$$

The proposed algorithm is depicted in Figure 1.

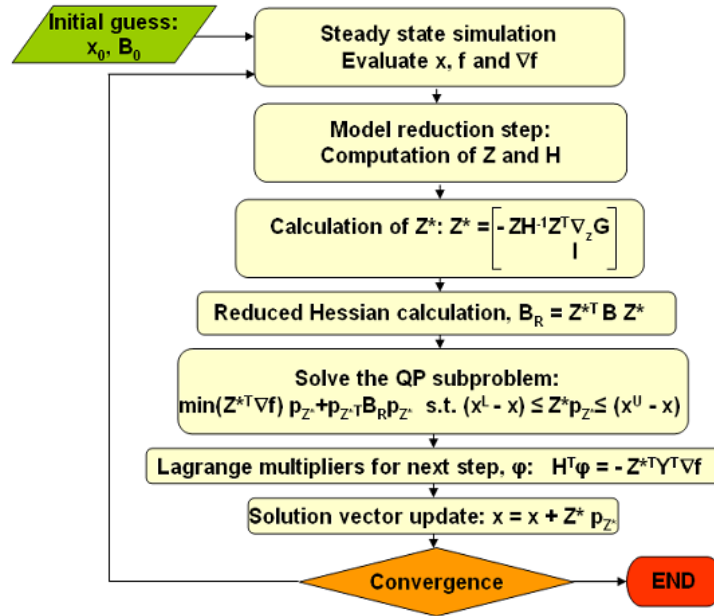


Figure 1: The proposed algorithm.

Case Study I: Optimization of a tubular reactor with 3 independent variable

To illustrate the behaviour of the proposed algorithm, we have chosen the case of the optimization of a tubular reactor with axial dispersion, where an elementary first order irreversible exothermic reaction takes place ($A \rightarrow B$), as it is known to exhibit a rich parametric behaviour.

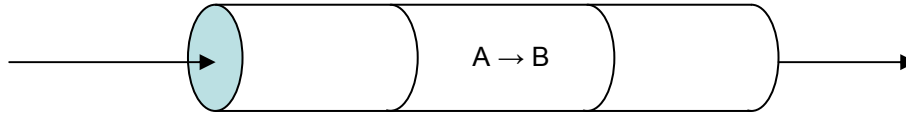


Figure 2: Schematic of a the tubular reactor, with 3 cooling zones

The model of the tubular reactor consists of two Partial Differential Equations, which for steady state become [5]:

$$\frac{1}{Pe_1} \frac{\partial^2 \mathbf{x}_1}{\partial y^2} - \frac{\partial \mathbf{x}_1}{\partial y} + Da(1 - \mathbf{x}_1) \exp\left(\frac{\mathbf{x}_2}{1 + \mathbf{x}_2/\gamma}\right) = 0 \quad (16)$$

$$\frac{1}{LePe_2} \frac{\partial^2 \mathbf{x}_2}{\partial y^2} - \frac{1}{Le} \frac{\partial \mathbf{x}_2}{\partial y} - \frac{\beta}{Le} \mathbf{x}_2 + \frac{C}{Le} Da(1 - \mathbf{x}_1) \exp\left(\frac{\mathbf{x}_2}{1 + \mathbf{x}_2/\gamma}\right) + \frac{\beta \mathbf{x}_{2w}}{Le} = 0 \quad (17)$$

Here \mathbf{x}_1 and \mathbf{x}_2 are the dimensionless concentrations and temperatures and \mathbf{x}_{2w} is the dimensionless wall temperature and y the dimensionless longitudinal coordinate. We consider three cooling zones on the jacket of the reactor, each one of which has a temperature that can be controlled independently. In this case, the dimensionless temperature at the jacket of the reactor is given by Eq. (18)

$$\mathbf{x}_{2w}(y) = \sum_{j=1}^3 [H(y - y_{j-1}) - H(y - y_j)] \mathbf{x}_{2wj}, \quad y_1 = 1/3, \quad y_2 = 2/3, \quad y_3 = 1 \quad (18)$$

All the parameters are described in [5]. The boundary conditions chosen for this case are:

$$\frac{\partial \mathbf{x}_1}{\partial y} - Pe_1 \mathbf{x}_1 = 0, \quad \frac{\partial \mathbf{x}_2}{\partial y} - Pe_2 \mathbf{x}_2 = 0 \quad \text{at } y = 0, \quad \frac{\partial \mathbf{x}_1}{\partial y} = 0, \quad \frac{\partial \mathbf{x}_2}{\partial y} = 0 \quad \text{at } y = 1 \quad (19)$$

The values chosen for the rest of the parameters are $Le = 1.0$, $Pe_1 = Pe_2 = 5.0$, $\gamma = 20.0$, $\beta = 1.50$, $C = 12.0$. The Finite Difference (FD) method was used for the discretization of the equations at hand and a number of nodes of 250 was chosen, leading to 500 algebraic equations.

The objective is to maximize the dimensionless concentration, x_1 , at the output of the reactor with respect to the values of the dimensionless temperatures of the 3 cooling zones and subject to Eqs. (16), (17) and (19) being satisfied.

The termination criterion for the algorithm was $\|z^* p_2\| \leq 10^{-6}$ and the size of the dominant subspace chosen was $m = 10$. The lower bounds for all variables were 0, the upper bounds were 1 for x_1 , 8 for x_2 and 4 for x_{2w} . The simulator of the tubular reactor was treated as a black-box simulator and all Jacobians and Hessians were calculated by employing numerical directional perturbations. Convergence was achieved in 18 iterations and the optimal point found is $x_{2w,1} = 2.4826$, $x_{2w,2} = 0.52539$, $x_{2w,3} = 4.0000$. For those values of the independent variable, a yield of 99.868% is achieved. In Fig. 3 the concentration and temperature profiles at the optimum for the reactor and for the cooling zones are presented.

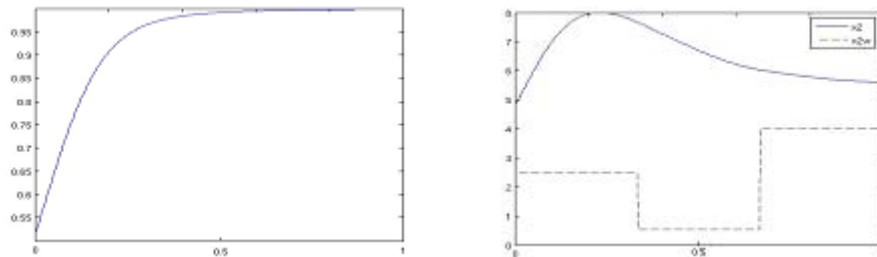


Figure 3: Solution profiles at the optimum point for dimensionless concentration (a) and dimensionless temperature (b) both for the reactor and for the 3 cooling zones.

Case Study II: Optimization of a Counter Flow Jet Reactor simulated using MPSalsa

To illustrate the efficiency of the proposed algorithm in handling large scale optimization problems using input/output simulators, we have applied it for the optimization of a Counter Flow Jet Reactor used for the decomposition studies of tertiary butyl arsine (TBA). The model of the reactor was setup using MPSalsa [7], which is state-of-the-art massively parallel CFD software, developed at SANDIA national laboratories. It implements the Finite Element Method, employing unstructured meshes and inexact Newton methods with iterative linear solvers. In this formulation, MPSalsa was used by our in-house developed optimization code as a black box.

A schematic of the reactor, as well as the formulation of the model and the pathways for the decomposition of TBA is presented in Fig. 4. The model considered is a 2D symmetric one. The momentum equation is:

$$\frac{\partial \mathbf{U}}{\partial t} = \rho(\mathbf{U} \cdot \nabla) \mathbf{U} - \nabla \cdot \boldsymbol{\tau} - g \rho \mathbf{e}_g \quad (20)$$

where \mathbf{U} is the velocity vector, ρ is the mixture density, which is a function of temperature and composition, g is the gravity vector and $\boldsymbol{\tau}$ is the stress tensor, which for a Newtonian fluid can be calculated from Eq. (21):

$$\boldsymbol{\tau} = -p\mathbf{I} - \frac{2}{3}\mu(\nabla \cdot \mathbf{U})\mathbf{I} + \mu(\nabla \mathbf{U} + \nabla \mathbf{U}^T) \quad (21)$$

where p is the operating pressure, μ is the mixture viscosity and \mathbf{I} is the unitary tensor. The total mass equation is

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (\rho \mathbf{U}) \quad (22)$$

The energy balance is:

$$\frac{\partial T}{\partial t} = \rho C_p (\mathbf{U} \cdot \nabla) T - \nabla \cdot (\kappa \nabla T) \quad (23)$$

where T is the temperature, C_p is the mixture heat capacity and κ is the thermal conductivity. The mass conservation equation for species i , $i = 1, \dots, N_c - 1$, N_c being the number of components of the mixture.

The model of the reactor consists of 19040 dependent variables (velocities, temperatures, pressures and concentrations for each point of the mesh) and 1 degree of freedom: the velocity of the upper stream.

The objective in this case is to maximize the production of AsH, thus maximizing the decomposition of TBA, while minimizing the production of the highly toxic AsH₃.

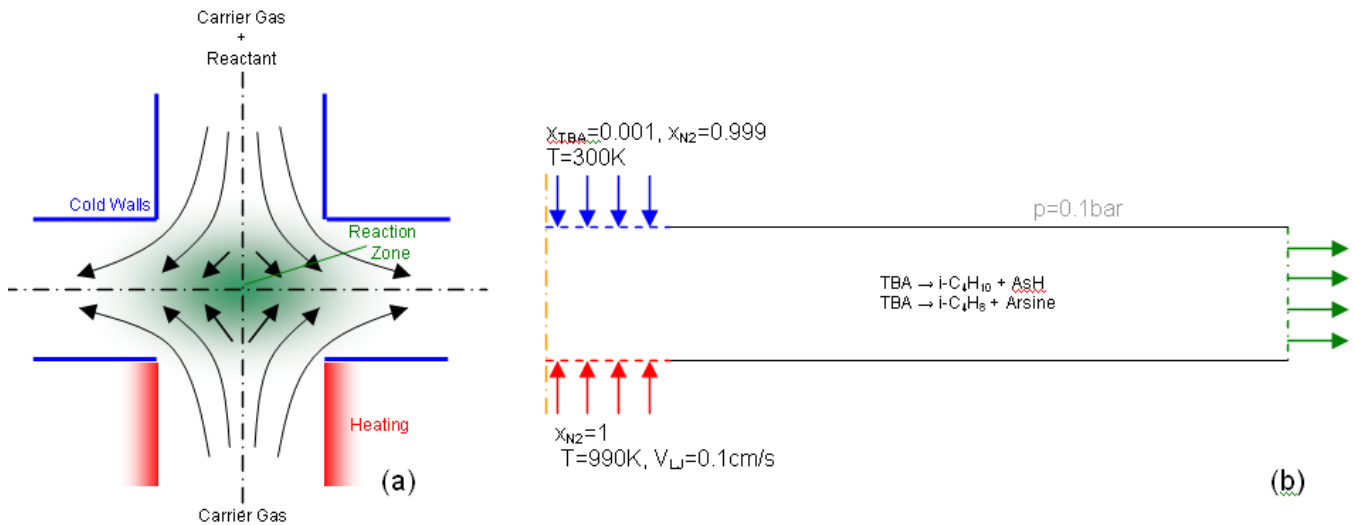


Figure 4: Schematic of the Counter Flow Jet Reactor (a) and formulation of the model (b).

The size of the dominant subspace chosen was 12. Convergence was achieved in 4 iterations and the optimal velocity found was 1.348cm/s, which results to a yield of 78.85%. Fig. 5 shows the temperature, velocity, pressure and concentration profiles at the optimum.

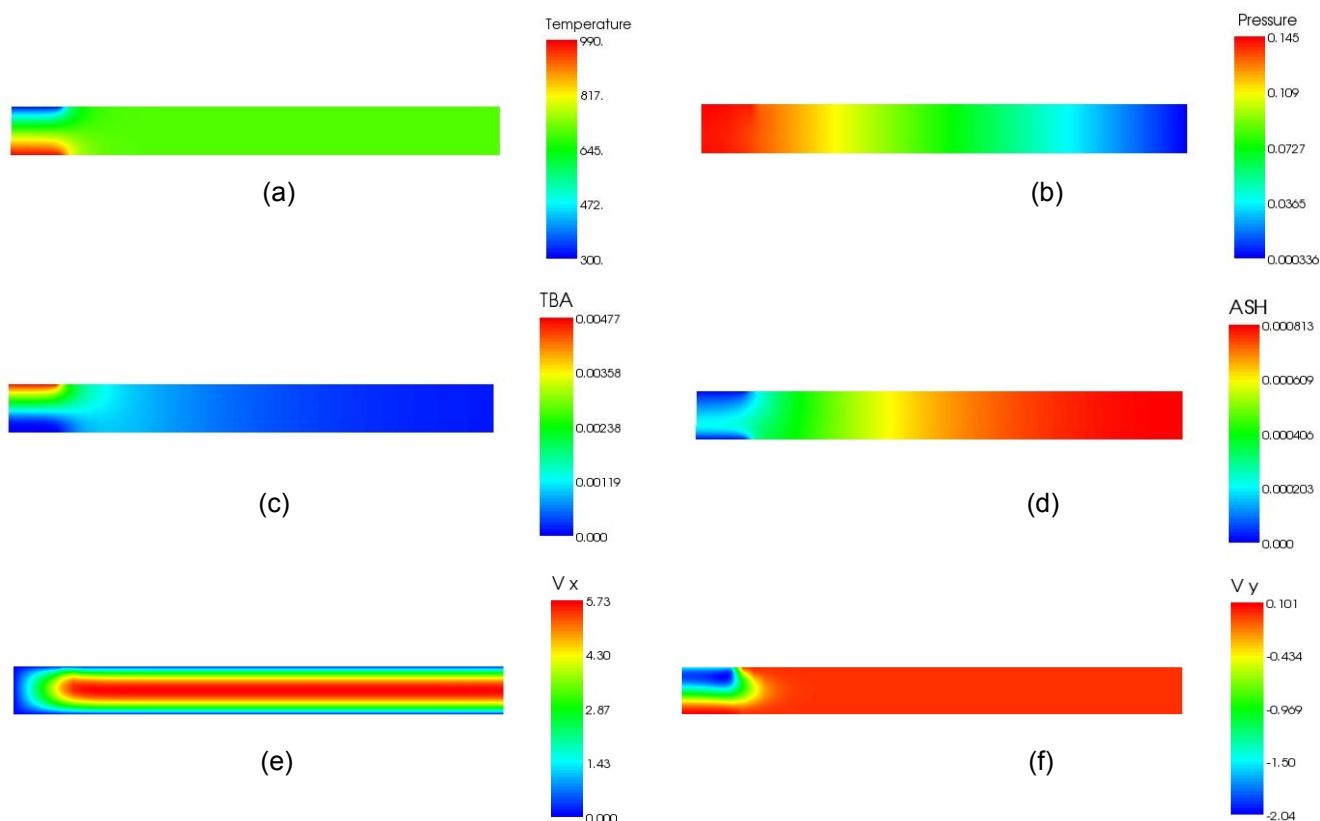


Figure 5: Schematic Temperature (a), pressure (b), TBA mass fraction (c), AsH mass fraction (d), and velocity (e and f) of the Counter reactor at the optimum point.

Acknowledgements

The financial support from the EU program CONNECT (COOP-CT-2006-031638) is gratefully acknowledged.

References

- [1] T.F. Edgar, D.M. Himmelblau, 1988, Optimization of Chemical Processes, McGraw-Hill, NY
- [2] C. Blum and A Roli, 2003, Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, ACM Computing Surveys, 35: 268-308.
- [3] E. Luna-Ortiz and C. Theodoropoulos, 2005, An input/output model reduction-based optimization scheme for large-scale systems, Multiscale Modeling & Simulation 4: 691-708.
- [4] C. Theodoropoulos and E. Luna-Ortiz, 2006, "A Reduced Input/Output Dynamic Optimisation Method for Macroscopic and Microscopic Systems", in Model Reduction and Coarse-Graining Approaches for Multiscale Phenomena, A. Gorman, N. Kazantzis, I.G. Kevrekidis, H.C. Ottinger and C. theodoropoulos (eds) pp. 535-560.
- [5] K. F. Jensen and W. H. Ray, 1982, The Bifurcation Behavior of Tubular Reactors, Chemical Engineering Science 37: 199-222.
- [6] Safvi, S.A. and T.J. Mountziaris, "A new reactor for purely homogeneous kinetic studies of endothermic reactions", AIChE Journal, 1994. 40: 1535-1548.

- [7] Shadid J, Hutchinson S, Hennigan G, Moffat H, Devine K, Salinger AG, "Efficient parallel computation of unstructured finite element reacting flow solutions", *Parallel Computing* 1997. 23: 1307-1325.
- [8] L.T. Biegler, J. Nocedal and C. Schmidt, 1995, A Reduced Hessian Method for Large-Scale Constrained Optimization, *Siam Journal on Optimization* 5: 314-347.
- [9] HSL, A Collection of Fortran codes for large-scale scientific computation. See <http://www.hsl.rl.ac.uk>, (2007).