RESERVOIR MODEL OPTIMIZATION UNDER UNCERTAINTY

By

Sasanka Are, Paul Dostert, Bree Ettinger

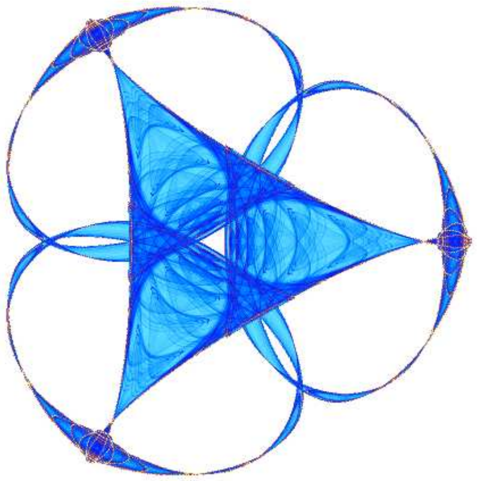
Juan Liu, Vadim Sokolov, Ang Wei

and

Klaus Wiegand (mentor)

IMA Preprint Series # 2133-4

(September 2006)



Reservoir Model Optimization under Uncertainty

IMA Mathematical Modeling Workshop

Final Report Team 4

Sasanka Are, University of Massachusetts

Paul Dostert, Texas A&M

Bree Ettinger, University of Georgia

Juan Liu, University of Florida

Vadim Sokolov, Northern Illinois University

Ang Wei, University of Delaware

Mentor: Klaus Wiegand, ExxonMobil URC

September 7, 2006

Abstract

Computerized reservoir simulation models are widely used in the industry to forecast the behavior of hydrocarbon reservoirs and connected surface facilities over long production periods. These simulation models are increasingly complex and costly to build and often use millions of individual cells in their discretization of the reservoir volume. Simulation processing time and memory requirements increase constantly and even the utilization of ever faster computers cannot stem the growth of simulation turnaround time.

On the other hand, decision makers in reservoir management need to quickly assess the risks associated with a certain model and production strategy and need to come up with high/low scenarios for Net Present Value (NPV) and the likelihood of these scenarios. To achieve reduced turnaround time in this difficult environment, reservoir engineers and applied mathematicians explore ways to employ optimization techniques that use surrogate models (i.e. a response surface) to perform these tasks. The costly simulation model is used to seed the design space and to assist with local refinement of the surrogate model.

In this report we summarize our findings from the IMA Math Modeling Workshop, in which we examined different algorithms to build surrogate models for a simple oil reservoir. Besides estimating NPV for certain high/low scenarios, we used surrogate models to find optimal producer well locations and to perform simple history matching of a 2D permeability field.

1 Introduction

1.1 Reservoir Simulation and Darcy's Law

1.1.1 Reservoir Simulation

Reservoir simulation models the flow of fluids through porous rock space. The term *fluid* is to be understood in the general sense, specifically this can be a combination of liquid hydrocarbons, water, and vapor. The movement of the fluids through the rock can be described using convection/diffusion equations that are discretized using finite volumes, finite differences, or finite elements. Boundary conditions are supplied in two forms:

1. No-flow Neumann conditions at the boundary of the reservoir
2. Mixed boundary conditions corresponding to injection and production wells

The boundary conditions introduced by the well-terms are usually time-dependent to mimic a certain injection/production history or development strategy. One of the many challenges in reservoir simulation is to optimize these strategies in such a way that the operator of a field maximizes the NPV over time without violating any constraints imposed by environmental regulations, operating limits, or business contracts.

1.1.2 Darcy's Law

In the mid-nineteenth century, Henry Darcy established a relationship for the volumetric flow rate **Q** through sample of porous media:

$$Q = -\frac{KA}{\mu} * \frac{\Delta p}{L}$$

The relationship states that the volumetric flowrate is proportional to the area through which the fluid passes, the pressure drop ΔP, and a hydraulic conductivity coefficient K which we call *permeability*. The flow rate is inversely proportional to the viscosity μ of the fluid and the length of the sample. The differential form of Darcy's law in one dimension (say: in the x-direction) is:

$$v = \frac{Q}{A} = -\frac{K}{\mu} \cdot \frac{\partial p}{\partial x}$$

where **v** denotes the (superficial) phase velocity. In three dimensions, Darcy's law becomes:

$$\vec{v} = -\frac{K}{\mu} \nabla p$$

where **K** is a symmetric, positive definite tensor. Combining Darcy's law with the divergence theorem, one arrives at a general equation for *single phase flow* in three dimensions:

$$\nabla \cdot \left[\frac{\rho \cdot K}{\mu} (\nabla p - \rho g \nabla D) \right] = \frac{\partial(\phi\rho)}{\partial t}$$

Here, ρ is the density of the fluid, ϕ is the porosity of the rock and g is the gravitational constant. The equation states that the amount of fluid entering a particular region is identical to the amount accumulated plus the fluid leaving the region. Hence, mass is being conserved. Accumulation (in this equation) is due to porosity changes and the assumption that the pore space of the reservoir rock is always filled with fluid.

1.2 Mathematical Formulation for Simulator

1.2.1 Some Terms

During the workshop, our team used a simple 2-phase reservoir simulator written in MATLAB. The phases were assumed to be liquid hydrocarbons with dissolved gas and water. The simulator used a simple, backward-in-time finite difference equation based on the principle of *volume balance*. Before we describe this equation, we briefly introduce some terms:

- **Components** vs. **Phases**
  Components such as **oil**, **gas** and **water** in the simple *black-oil* fluid model represent mass which could be measured in kg or similar. However, reservoir engineers tend to use the term *moles* when describing mass and contrary to what a chemist might think, the units used are barrels and scf (for gas). **Phases** describe states of fluid regimes with (assumed) homogeneous properties. Phases contain a mix of components, but not every component disssolves in all phases. For instance, oil does not dissolve in water. For phases, volume is recorded and the total fluid volume is the volume of all phases combined (which should match the pore volume of the rock). Phases are more or less compressible. Therefore, if the total fluid volume does not match the pore volume for a given cell, the cell's pressure has to be adjusted to either compress or decompress the fluid.
- **Molar Flow** vs. **Phase Flow**
  Molar flow is the flow of material and is usually measured in barrels or scf. Phase flow is always volumetric. Conversion factors and molar densities are used inside a simulator to convert back and forth between these quantities. Based on the pressure inside the reservoir, the volumetric flow rates and the volume transported may represent a larger amount of material than they would under surface conditions.

1.2.2 Volume Balance Equation

Our MATLAB simulator was based on the industrial-standard Volume Balance Equation. This equation establishes a relationship between the pore volume of a cell and the total fluid volume within the cell. The basic idea is that the total fluid volume will change primarily due to mass changes induced by molar flow in and out of the cell. This change is compensated by an adjustment of the fluid volume due to pressure changes inside the cell, and an adjustment of the pore volume due to a pressure-dependent rock-porosity. In our case, the rock compressibility was assumed to be zero, so the only adjustment came from fluid volume changes due to pressure change. The equation itself is implicit in pressure but not with respect to the mass transported or the resulting phase saturations. The equation is therefore called the *IMPES* equation (implicit in pressure and explicit in saturation). Here is the equation:

$$\left[\frac{\partial V_p}{\partial p} - \frac{\partial V_t}{\partial p}\right]_i^n \delta P_i + \Delta t \cdot \sum_j \sum_m \sum_{k=i,j} \left[\frac{\partial V_t}{\partial N_m} \frac{\partial U_{m,i\to j}}{\partial p}\right]_k^n \cdot \delta P_k = [V_t - V_p]_i^n - \Delta t \cdot \sum_j \sum_m \left[\frac{\partial V_t}{\partial N_m}\right]_i^n \cdot U_{m,i\to j}$$

1.3 Simulator Implementation

1.3.1 Solving the Volume Balance Equation

One way to solve the volume balance equation is to simply solve the linear system, calculate the distribution of components among the phases and the resulting phase volumes, and (after a sanity check) move on to the next timestep. Since the calculation of the derivatives in the volume balance equation introduces nonlinear terms, there will be a smaller or larger *volume discrepancy* at the beginning of the new timestep, that is, the term $[V_t - V_p]_i^n$ will not be zero, and this explains why the term is part of the equation in the first place.

Another way to proceed is to solve the linear system, upate the pressures and fluid volumes and iterate in a quasi Newton scheme until the largest individual pressure change falls below a certain minimum. During the iteration, fluxes on the rhs and moles (mass) per cell are held constant. Unless we leave the stable region, the iteration should converge (that is, the rhs and solution vector should approach the zero vector) and the term $[V_t - V_p]_i^n$ should vanish. We chose this scheme to solve the volume balance equation in our experiments.

1.3.2 Updating Properties

Once the volume balance equation is solved, new fluxes are calculated based on the updated pressures and Darcy's law. The new fluxes multiplied by the timestep-size determine how much material is transported in and out of the cells, which in turn allows us to update the moles of each component. The last step is to calculate the distribution of the components within the existing phases and the the resulting phase volumes for each cell.

1.3.3 Timestep Control

We used a very simple scheme to control the timestep size in our experiments. We assumed a target of 3 Newton iterations for solving the volume balance equation and increased the timestep size if we needed ≤ 3, and decreased it when we needed more. In case the linear solver or the Newton iteration did not converge, we restarted the timestep with a smaller Δt.

2 Workshop Setup

2.1 Workshop Tasks

During the IMA Math Modeling Workshop, our team had the task to explore the use of different response surface models for reservoir simulation and perform a qualitative analysis of the developed models. Specifically, our research was broken down into the following parts:

1. Creation of response surface models for a given reservoir using a simplified black-oil reservoir simulator to seed the design space.
2. Prediction of the reservoir's NPV for various parameter combinations using the response surfaces.
3. Optimal well placement using the response surfaces.
4. History Matching

2.2 Reservoir Models

2.2.1 Model Parameters

Of the many parameters that define the behavior of a hydrocarbon reservoir, *absolute rock permeability* and *relative phase permeability* are among the most important. As we have seen in Darcy's law, do these two parameters (and the viscosity of the fluid) control the ability or readiness of fluid to pass through the reservoir rock. Geological features such as faults, channels and layers all have a strong impact on the permeability distribution, and their effects can therefore be understood as permeability-induced effects. Porosity, the fraction that defines how much of the rock's bulk space is filled with fluid, also plays an important role since it ultimately defines how much fluid is present - although the recovery factor defines how much we are able to extract. Porosity therefore is important when we include the estimated reserves into our NPV calculation. In our research, however, we defined NPV to be the amount of fluid recovered over a relatively short interval (up to 10 years using only two producer wells) and therefore, porosity was expected not to be a controling factor. We actually used this to test the ability of our response surface models to filter out less important factors. We also need to mention that we ignored any correlation between porosity and permeability in our experiments.

Based on what was said above and the time constraints under which we had to operate, we decided to vary the folowing parameters/decision variables in our experiments:

Parameters/Factors

- Absolute Permeability

  Absolute permeability was varied either by defining constant values for certain subdomains of the reservoir or by defining a distribution function over the whole domain.

- Relative Phase Permeability

Relative phase permeability (physically) is a function of phase saturation and we wanted to keep our model as close to physical reality as possible. We therefore parameterized the function that calculated relative permeability and varied the parameter by region/subdomain.

- Porosity

  Porosity was varied by defining constant values for each subdomain.

Decision Variables

We only used one set of decision variables in our experiments due to time constraints.

- Well Location

  Holding all other parameters/factors constant, the location of up to two producer wells was changed in order to optimize the NPV of a given field.

Wells

The wells used in our experiments were *potential wells*. These wells have a bottomhole pressure (BHP) specified that translates into a Dirichlet boundary condition for the mathematical model of the reservoir. The specified BHP for our wells was always smaller than the reservoir pressure, and therefore the wells produced fluids. We did not model any injection due to the time-constraints we experienced.

2.2.2 Model Realization

Different models were used during our studies. All models were 2-dimensional, of rectangular shape, and used a simple structured grid. In the following, we give an overview of the different models and how they were used.

The **first** model was a reservoir, subdivided into three domains, with porosity and absolute rock permeability defined by region. Each region contained one producer well. The 3-domain model was used to build the first surrogate models and to test their ability to predict NPV based on parameter changes. We also wanted to see if the surrogate models were able to determine that porosity (for the *given scenario*) was not an important factor.

The **second** model we built was used to predict optimal well locations. Instead of using subdomains, absolute permeability and relative phase permeability were parameterized functions with fixed parameter values that resulted in a given distribution of these quantities inside the reservoir. The task was to find the optimal location for two producer wells and to forecast the total oil production over a given time period. We also included some nonpermeable rock-regions into the reservoir to make the task more challenging. These regions are depicted in the graph by the dark blue sections. It should be noted that simply placing both wells in the dark red spots will not work, since the pressure decline is too strong and the only drivers for oil recovery are the fluid compressibility and the initial reservoir pressure (since there are no injectors). Also, the closer a well is placed to the boundary, the less area is available for the well to draw fluid.

The **last** model we studied was used to perform a history match. Well locations and relative phase permeabilities were fixed while absolute permeability was an assumed function of two historic riverbeds, entering the reservoir at a certain position and direction and mixing with each other.

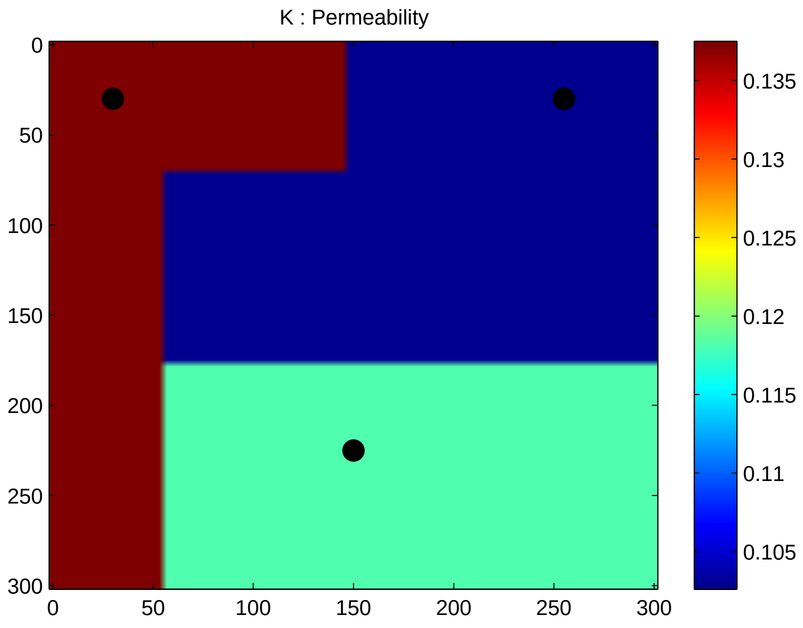


Figure 1: Reservoir divided into 3 Domains

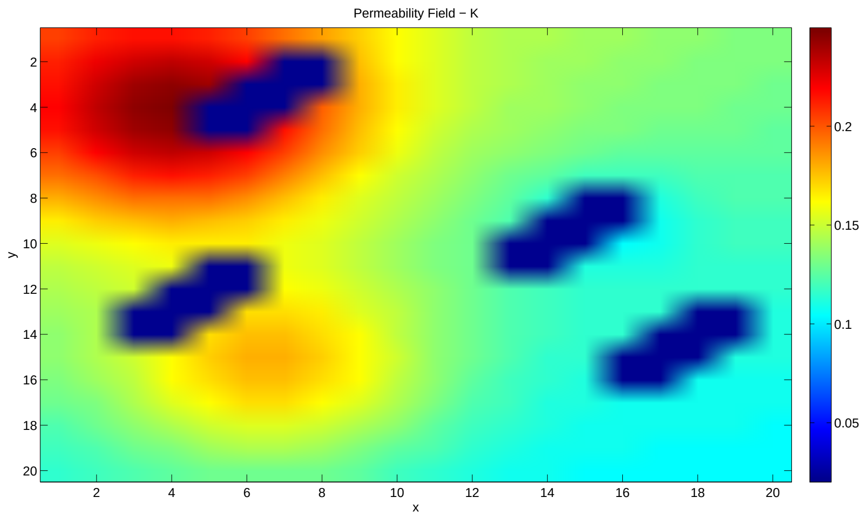


Figure 2: Reservoir for Well Placement Problem

The permeability distribution for different positions and angles was quite different and the task was to history match the distribution based on 360 production rate measurements for each of the two producer wells. Fig. 3 shows two of the many possible distributions.

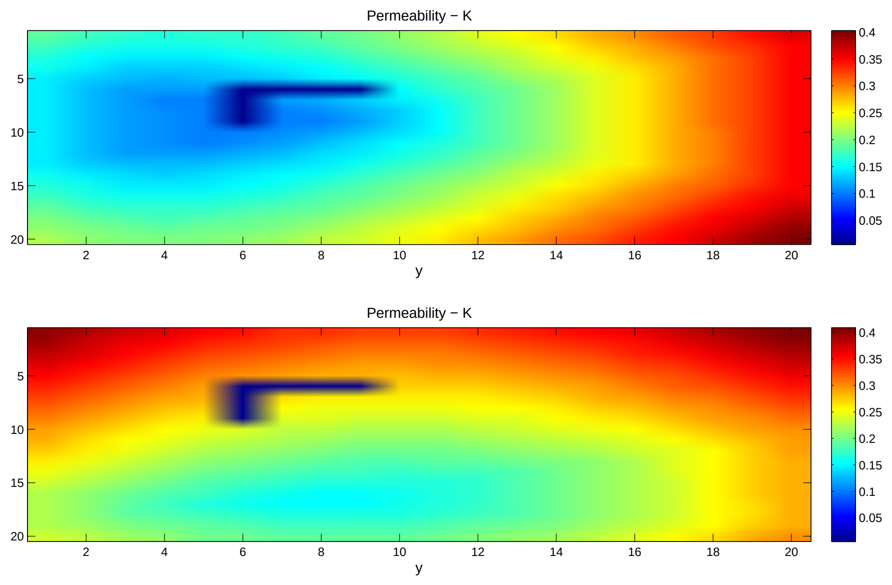


Figure 3: Permeability Distributons for History Match Problem

2.3 Simulation Environment

As stated, our team used a simplified 2-phase, 2-component black oil reservoir simulator implemented in MATLAB. The simulation model was restricted to 2 dimensions; capillary and gravitational effects were ignored. The simulation engine used a backward-in-time, implicit finite difference method to solve for pressure changes during a timestep based on volume balance. Primary variables were pressures and component moles in each cell. Phase saturations were calculated at the end of each timestep based on phase volumes, which in turn were calculated by a property package based on the primary variables. The model was initialized by specifying pressures and phase saturations in each cell and backing out the component moles using some initialization code. An external driver was constructed to run up to 100 individual simulations using different input parameter sets, and to capture the results needed to build the surrogate models. The driver was then distributed over up to 8 machines on the campus in order to parallelize the simulations.

3 Investigated Response Surface Models

During the workshop we investigated 3 different candidates for the construction of response surface models:

1. Linear Regression Models
2. Spline Models
3. Neural Network Models

Each of the three models had to complete all three tasks, however, since the spline code was restricted to bivariate splines, we had to tweak the tasks for this particular model. In order to accomplish the first task, a binning technique was used to group parameters. For the second task, the wells were assumed to be positioned on a subset of the reservoir space by using a single parameter to position each well within the subspace. For the third task, where we had 4 parameters (starting position and angle for two historic riverbeds), a special distribution function was written that depended on two variables only.

In the following we discuss the different approaches taken as well as the achieved results for each of the three posed problems.

3.1 Regression Model

3.1.1 Regresion Model Design

The generic form of the regression model is

$$y = f(x_1, ..., x_k) + \varepsilon$$

where y is the explained variable, x_i are explanatory variables, $\varepsilon \sim N(0, \sigma^2)$ and f is a polynomial. The main goal is to find the best f which maximizes the coefficient of determination R^2 and minimizes the unbiased estimator of σ^2, s^2. The procedure is to start with a complex candidate model based on the physical meaning of the parameters x_i and y, using the goodness-of-fit test and the significance of coefficients test, the number of parameters can be reduced. The process is repeated until the desired f is obtained.

3.1.2 Model 1: Oil Production depends on Parameters

Model Estimate

There are three domains, one producer in each domain at a fixed location. The goal is to find the relationship between the total oil production (for a given time period) and the properties of the field. We assume all three factors are uniform in three regions. Therefore, the production level $P = f(\phi, K, \gamma) + \varepsilon$, is a function of three variables, where ϕ and K are vectors of corresponding values for porosity and permeability in the 3 regions, γ is a vector of parameters for the parameterized relative phase permeablity, f is a polynomial and $\varepsilon \sim N(0, \sigma^2)$. After an initial test which showed that the coefficient for porosity was extremely small, a second candidate model was built, containing linear terms of K and γ, and the cross terms between them. Using the goodness-of-fit test and the significance of coefficients test, we were able to reduce the number of parameters, obtaining the following model:

$$P = x^T\beta + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

where $x = [1, K_1,\ K_2,\ K_3,\ K_1\gamma_1,\ K_2\gamma_2,\ K_3\gamma_3]^T$, and $\beta = [\beta_0\ \beta_1\ \cdots\ \beta_6]^T$.

Parameter estimates($\times 10^4$) (standard errors in parentheses):

$$\hat{\beta}_0 = -0.103(0.0052) \quad \hat{\beta}_1 = 32.758(0.7612) \quad \hat{\beta}_2 = 37.701(0.2756) \quad \hat{\beta}_3 = 49.461(0.3474)$$
$$\hat{\beta}_4 = -17.767(2.9850) \quad \hat{\beta}_5 = -28.145(0.6096) \quad \hat{\beta}_6 = -49.362(0.5390)$$

The variance of the error is $s^2 = 679.9703$, the coefficient of determination $R^2 = 1$. Fig. 4 demonstrates the fitness of the regression model.

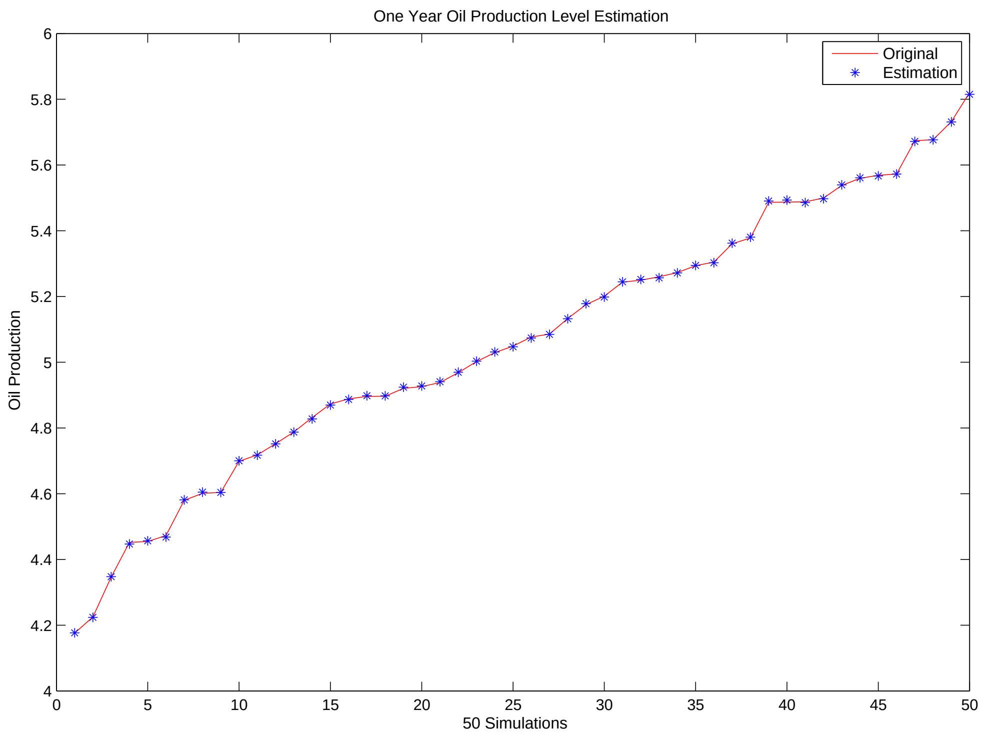


Figure 4: Fit of Regression Model for Task 1

Hypothesis Testing

- The overall regression is not statistically significant.

  H_0: All slopes in the model are zero
  H_1: Not H_0

  The test statistic is

$$F[k-1, n-k] = \frac{R^2/(k-1)}{(1-R^2)/(n-k)}$$

  In our problem $n = 50, k = 7$ and $R^2 = 1$, so the F ratio for this test is $F(6, 43) = 2.0583 \times 10^5$, which is lager than the 99% critical value of 7.1. We conclude that the data are inconsistent with the hypothesis that all of the slopes in the equation are zero. Therefore H_0 is rejected, the overall regression is statistically significant.

- For each β_i,$i = 0, ..., 6$, is equal to zero.

  $H_0 : \beta_i = 0$

$H_1 : \beta_i \neq 0$

The test statistic is

$$t[n-k] = \frac{\hat{\beta}_i}{\sqrt{Var\hat{\beta}_i}}$$

All the t values of the β_i's are larger than the 99% critical value of the t distribution (with 43 degrees of freedom), therefore we reject H_0. Hence, each β_i, $i = 0, ..., 6$ is significantly different than zero.

Model Implementation

- t value for K_i's are relatively large compared to others, which shows the importance of permeability in our problem.
- $\frac{\partial P}{\partial K_i}$'s are positive numbers,
  which means higher permeability within the reasonable range(0.05, 0.25) brings greater production.
- $\frac{\partial P}{\partial \gamma_i}$'s are negative numbers,
  which means higher γ_i values lower the production level. This is consistent with the fact that the parameterized version of the relative permeability function does indeed lower oil relative permeability for higher values of γ.

Model Prediction Capability

To test the model's predictive performance, we generated 20 new simulations and applied the model to predict the oil production level. Although the model needed to extrapolate some of the data, the prediction is still within 5% error. Fig. 5 shows the predictive performance of the model.

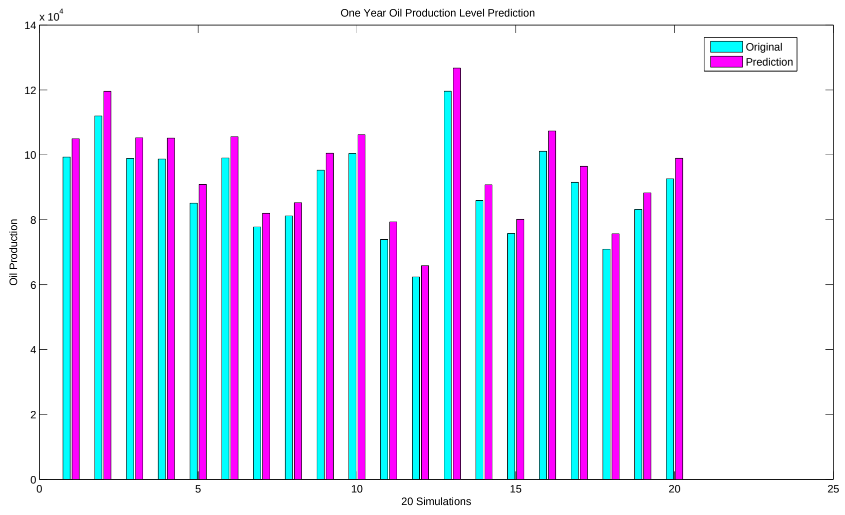


Figure 5: Regression Model Predictive Performnce

3.1.3 Model 2: Optimized Well Placement

Model Estimation

The goal of this model was to find the relationship between the oil production level and well locations. The explained variable is total production level of two producers P over a 10-year period. The explanatory variables are the locations of two producers (x_1, y_1) and (x_2, y_2). For this scenario, two factors strongly affect the production level, the closeness of the two producers and the distribution of the permeability over the field. Based on the first factor, one may expect that the two x_i's and two y_i's are dependent, but x_i and y_j should be independent. Taking into account the second factor, one can conclude that x_i and y_i are highly related. Combining all the information leads to the following model.

$$P = x^T\beta + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

Where $x = [1,\ x_1,\ y_1,\ y_1^2,\ y_2,\ x_1x_2,\ y_1y_2,\ y_1^3,\ y_2x_2^2,\ y_1^2y_2,\ y_1y_2^2,\ x_2y_2^2,\ x_1^2y_1,\ x_1y_1^2,\ y_1^4,\ x_1x_2^3,\ y_1^2y_2^2]^T$, and $\beta = [\beta_0\ \beta_1\ \cdots\ \beta_{15}]^T$. Parameter estimates (Standard Errors in Parentheses):

$$\begin{aligned}
&\hat\beta_0 = 75916.58(8.5264) \quad \hat\beta_1 = -5337.34(-1.3872) \quad \hat\beta_2 = 129317.91(2.8295) \quad \hat\beta_3 = -24689.03(-3.0159)\\
&\hat\beta_4 = 363.74(1.3030) \quad \hat\beta_5 = 6566.42(2.9998) \quad \hat\beta_6 = 1620.67(2.8263) \quad \hat\beta_7 = -88.81(-3.6217)\\
&\hat\beta_8 = -354.92(-2.7422) \quad \hat\beta_9 = -385.75(-3.2697) \quad \hat\beta_{10} = 80.83(2.4471) \quad \hat\beta_{11} = -109.07(-5.3127)\\
&\hat\beta_{12} = 135.54(6.6454) \quad \hat\beta_{13} = -35.73(-2.5907) \quad \hat\beta_{14} = -1.14(-1.4629) \quad \hat\beta_{15} = 17.49(2.7705)
\end{aligned}$$

For this model, n=50, k=16, $R^2 = 0.8956$, F(15,34)=19.4513. The standard deviation of the error is 4.7401×10^3. See Fig. 6 for the fitness of the regression model.

Model Prediction Capability

Twenty new simulations were generated and used to test the model performance. Although the data values are out of the range for which we built the model, the prediction was within 9.6% error. See Fig. 7 for the model's prediction capability.

Model Optimization

The optimal well locations can be found by optimizing the production function. The optimal well locations are $(x_1, y_1) = (1, 5)$ and $(x_2, y_2) = (5, 10)$, which produces 1.07×10^5 barrels of oil in 10 years.

3.1.4 History Match

Model Description

For this model we wanted to estimate the permeability pattern that best matches a given production history for two producer wells over a period of 10 years. With the initial four x_i values for the starting and ending positions of two historic riverbeds, we can obtain the error values for different patterns. Then we can optimize the model, and find where the minimum error occurs.

Model Estimation

According to the routine of model designing, the following model can be derived respected to the data set:

$$R = X\beta + \varepsilon$$

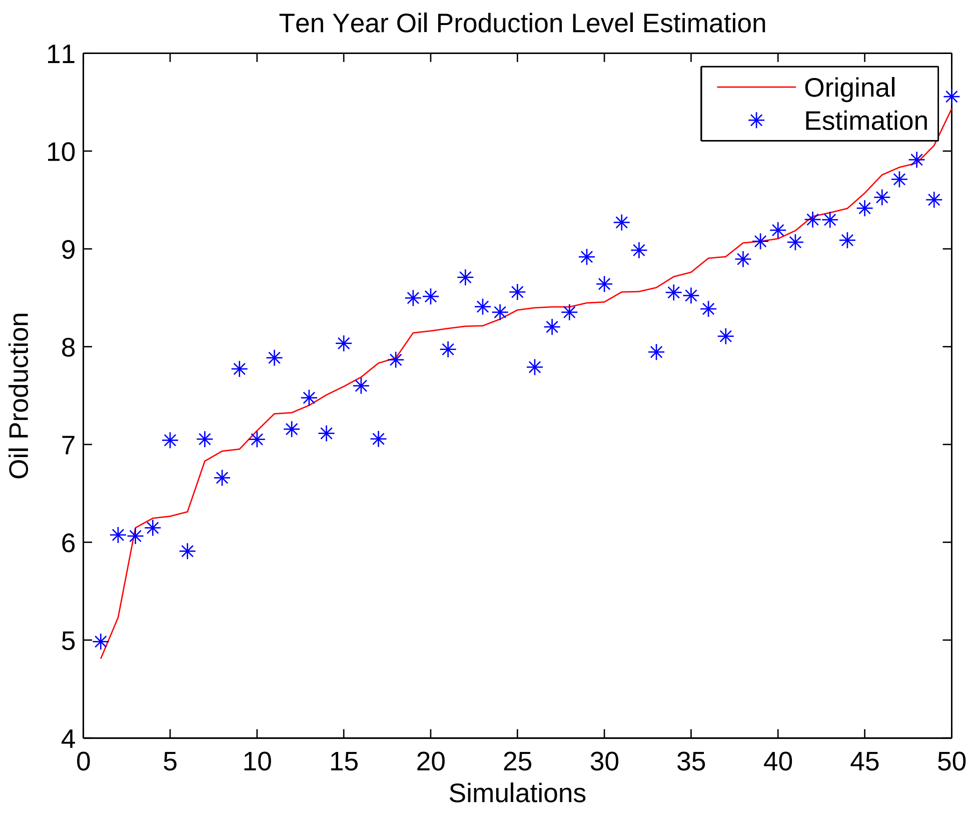


Figure 6: Fitness of Regression Model for Well Placement Problem

where $\beta = [\beta_0, \beta_1, ..., \beta_{34}]^T$ and

$$\begin{aligned} X = \quad & [x_3,\ x_4,\ x_1,\ x_2,\ x_3^2,\ x_4^2,\ x_2^2,\ x_3x_1,\ x_4x_2,\ x_1x_2, \\ & x_3^3,\ x_1^3,\ x_4^3,\ x_2^3,\ x_3x_1^2,\ x_1x_3^2,\ x_3x_4^2,\ x_4x_3^2,\ x_3^4,\ x_4^4,\ x_1^4, \\ & x_2^4,\ x_3x_1^3,\ x_1x_3^3,\ x_4x_2.^3,\ x_2x_4^3,\ x_3x_4^3,\ x_4x_3^3,\ x_2x_1^3,\ x_1x_2^3, \\ & x_3^2x_1^2,\ x_4^2x_2^2,\ x_3^2x_4^2,\ x_2^2x_1^2]^T \end{aligned}$$

For this model, n=50, k=34, the statistics are:

$$\begin{aligned} & F(33,16) = 97.024 > F_{26,33}, \quad min(|t_{\beta_i}|) = 1.231 > t_{26}(1-0.1) \\ & s^2 = 0.064, \quad R^2 = 0.995 \end{aligned}$$

Fig. 8 shows the estimations and real values.

Optimization

The error estimation model can be used to find the permeability distribution which minimizes the production error estimate. By minimizing the R function, one can obtain the following coordinates where the minimum error is 95:

$$x_1 = 11, \quad x_2 = 4, \quad x_3 = 6, \quad x4 = 8$$

Compared with the actual value, the error for model optimization is about 6%. It is an acceptable result due to the data-dependent model, see Fig. 9.

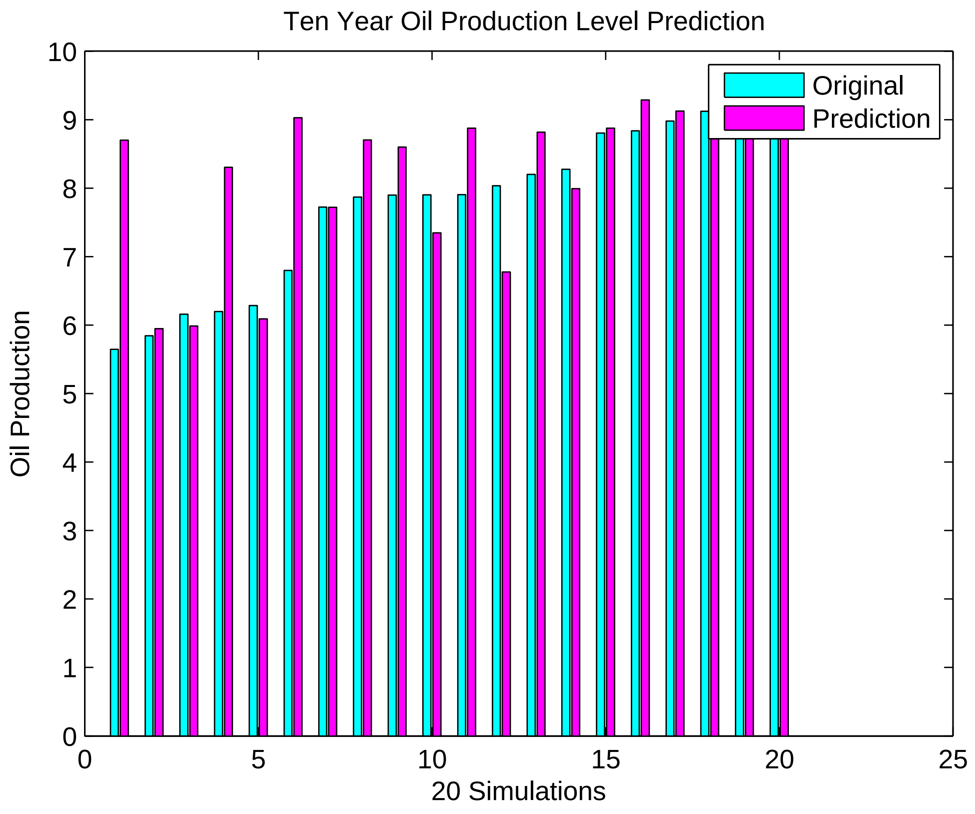


Figure 7: Prediction Performance for Well Placement Problem

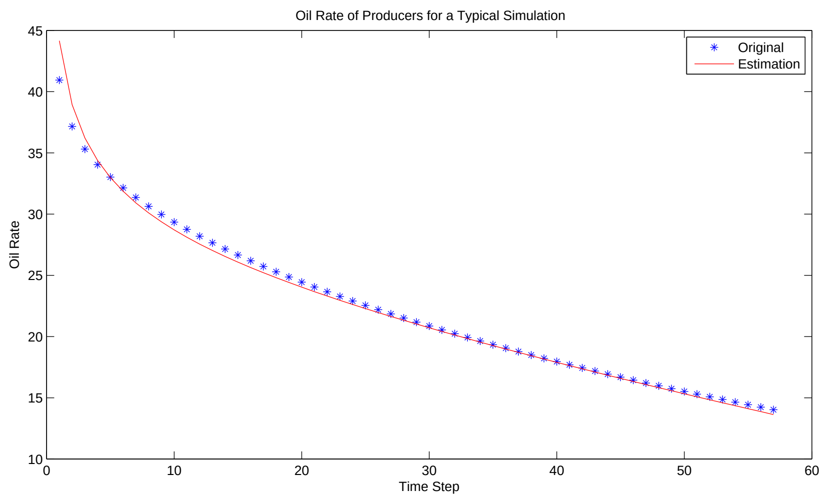


Figure 8: Model Estimation for History Match

3.1.5 Discussion

1. Data dependency: all three models were estimated using a limited set of data, which cannot range over the whole domain of the explanatory variables, so the models may not predict accurately for data outside the given range.

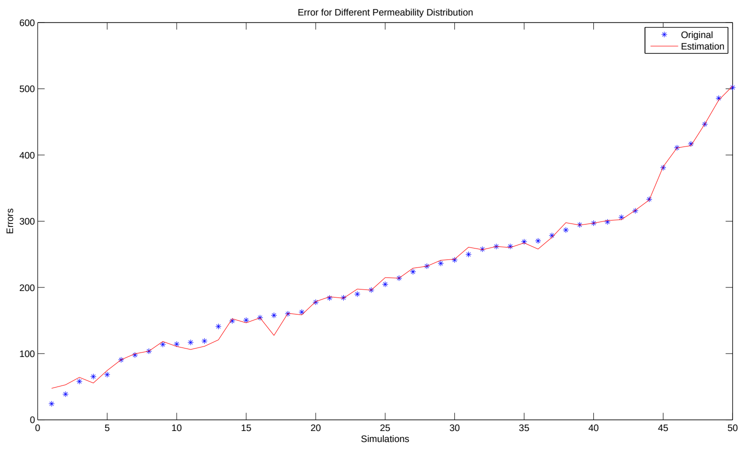


Figure 9: Model Prediction for History Match

2. Re-scaling: the range of the time value is $[0.01, 3650]$, i.e. the maximum value is 10^5 times the minimum value. Due to the limited computer capacity, re-scaling must be applied to t in order to increase the model stability and accuracy.

3.2 Model Analysis using Splines

In the previous subsection we presented a regression model with reasonable prediction capabilities. In this subsection we present a model based on bivariate splines. We consider a spline model approach because of the good interpolation properties of splines. With any stochastic/regression modeling, the number of simulations required is very high, whereas to generate a spline model requires a lot fewer simulations. Also the spline surface that models the simulator can be used to sample points for other models. The spline model presented here has only two dimensions but can be extended to multiple dimensions. The surface generated by the bivariate spline model is a good visual representation of the relation between different parameters.

3.2.1 Splines

Given $\triangle = \{T_i\}_{i=1}^N$ a triangulation of a polygonal domain Ω and integers $0 \le r \le d$, we define the space of polynomial *bivariate splines* of smoothness r and degree d to be the linear space

$$\mathcal{S}_d^r(\triangle) = \{s \in C^r(\Omega) \; : \; s|_{T_i} \in \mathcal{P}_d, \; i = 1, \dots, N\}$$

where

$$\mathcal{P}_d := \left\{ p(x,y) \;=\; \sum_{0 \le i,j \le d} c_{ij} x^i y^j \right\}$$

is the space of polynomials of degree $\le d$. We will use the minimal energy method to find a smooth bivariate spline to interpolate the data. We want to use the minimal energy method because it will produce good fit to the data without too many drastic changes in curvature. For any scattered data set $\{(x_i, y_i, f(x_i, y_i)), \; i = 1, \dots, N\}$ where all of the points are distinct and we take $\triangle$ to be a triangulation of the given data locations with vertices at the given data locations. Then let $\mathcal{S} := S_d^r(\triangle)$ be the spline space of degree d and smoothness r with $d \geq 3r+2$ over $\triangle$. Take

$$\Lambda(f) := \{s \in \mathcal{S} \ : \ s(x_i, y_i) = f(x_i, y_i), i = 1, \ldots, N\}.$$

and choose $s_f \in \Lambda(f)$ such that

$$E(s_f) := \min_{s \in \Lambda(f)} E(s)$$

where

$$E(s) := \sum_{t \in \triangle} \int_T \left(s_{xx}^2 + 2s_{xy}^2 + s_{yy}^2\right) dxdy. \tag{1}$$

we call s_f the minimal energy solution. For our tests we will take $r = 1$ and $d = 5$. For existence, uniqiness and convergence see [Awanou, Lai, Wenston].

3.2.2 Numerical Experiments

Since the bivariate spline model uses only two parameters, we reduce the dimensionality of the problem by making some further assumptions and introduce a process called binning. In our first run, we consider K the permeability and ϕ the porosity as the two parameters affecting the oil production of our simulator. To bin a given set of six variables (ϕ_1, ϕ_2, ϕ_3, K_1, K_2, K_3), we calculate ϕ_{min} the lowest porosity of a given reservoir, and ϕ_{max}, the highest porosity of the reservoir, and loft the surface by the point

$$\phi_{mid} = \phi_{min} + \frac{\phi_{max} - \phi_{min}}{2}.$$

The same process is done for K. Thus we only have two variables, ϕ and K. Fig. 10 below illustrates the process.

Recall that based on the physical meaning of the parameters ϕ, K, and γ, we guess that rock permeability and phase relative permeability play important roles in production. The left side of Fig. 11 shows the predicted linear relationship between K and total oil production(TOP). The right side of the figure shows that TOP is not effected by ϕ in this problem. Since ϕ does not affect the simulator oil production for this problem, we fix it and consider K and γ as parameters affecting the TOP. Fig. 12 shows the regression model's mixed term results.

To verify the model, we predict production values using the spline model considering only K and γ. When the points chosen for prediction lie in the surface, the spline model does very well. Since the K-γ interaction is fairly simple, it looks like we can make predictions by extrapolating the surface. Keep in mind that we would only want to do this for points sufficiently close to the surface, since we are using piecewise polynomials.

3.2.3 Optimization of well locations

For the first two optimization problems, we will consider well locations along two fixed curves. The first producer will lie on

$$f_1(x) = \frac{1}{1 + \exp(-16(1-x) + 8)}$$

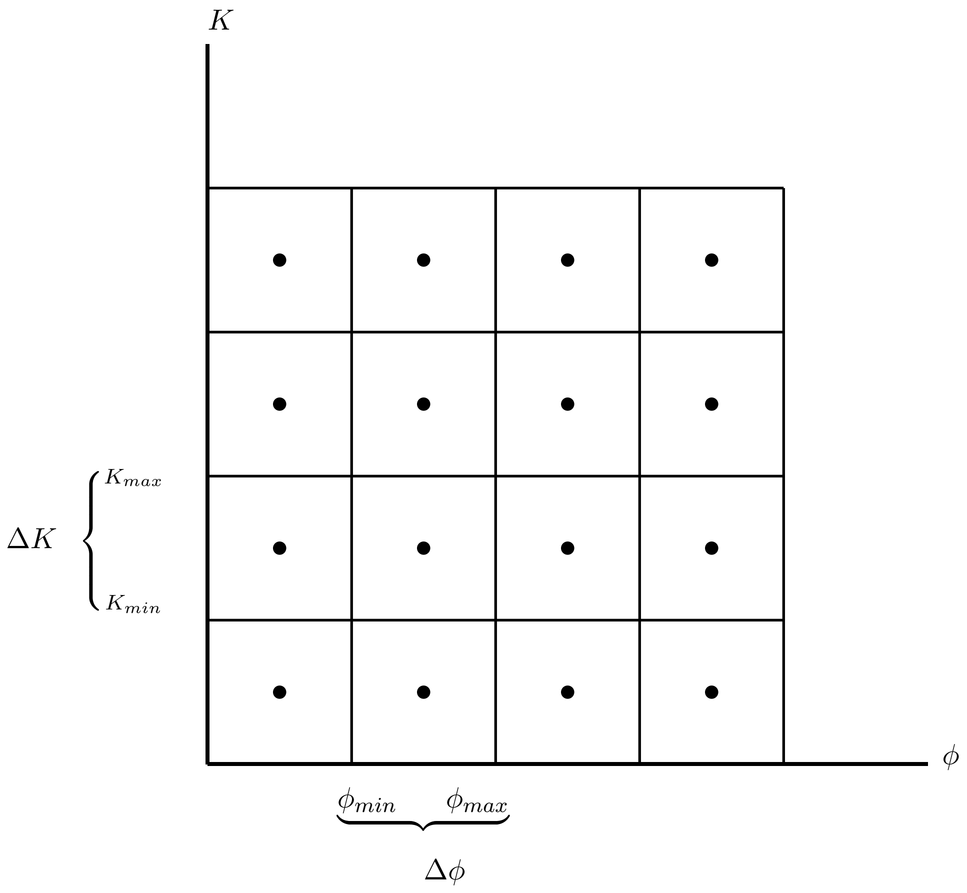


Figure 10: This figure demonstrates the binning process. The black points in the centers of the "bins" or squares are (ϕ_{mid}, K_{mid}). We loft the spline by the black center points.

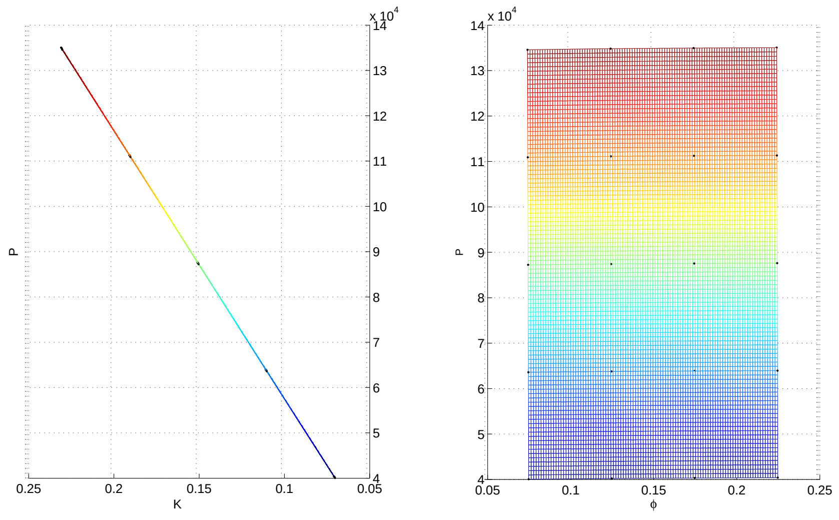


Figure 11: In the left plot we can see the linear relationship between K and total oil production. The right plot shows that ϕ does not affect the total oil production in our simulator.

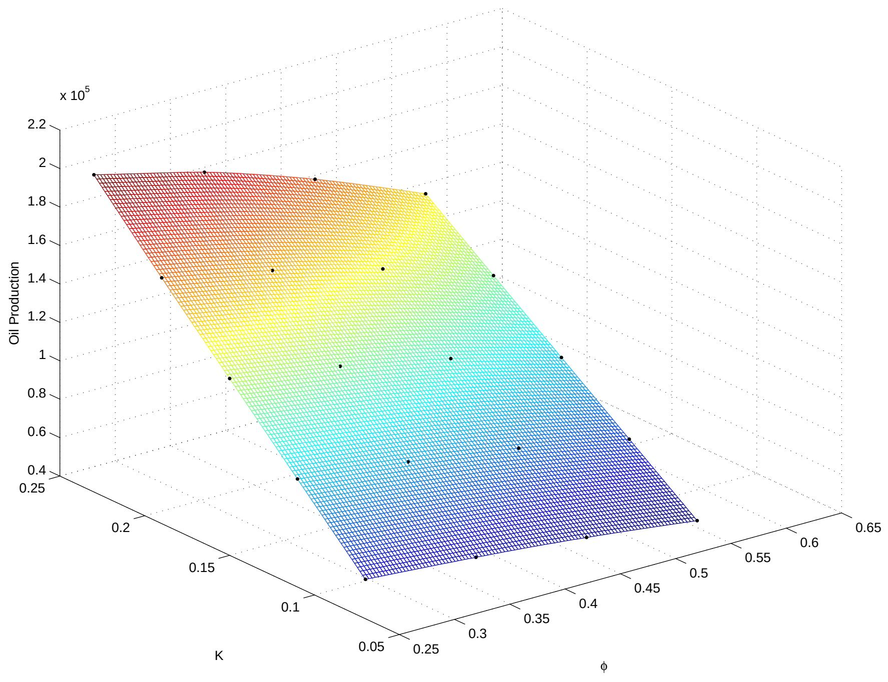


Figure 12: In the above plot you can see the slight arc in the surface cause by the relationship between K and γ.

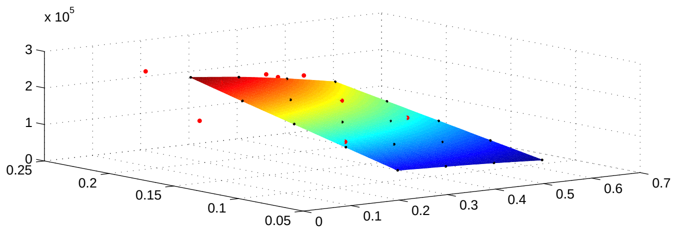


Figure 13: The red points are the points we are trying to predict, the black points are the knots for the surface. Those points that lie in the domain of the original surface are well predicted.

while the second producer will lie on the curve

$$f_2(x) = \frac{1}{1+\exp(-4.5(2x-1))}.$$

We will reduce the problem to two dimensions by optimizing x_1, (the x location for the first producer) and x_2 (the x location for the second producer). We choose x values randomly, evaluate them on f_1 and f_2, and then round them to get a whole number on our reservoir grid. In our first attempt we use 31 points to loft the surface. In Fig. 14, which depicts this model, the maximum error in production is 15,365 barrels (the relative error being 15%). The reason for the large relative error lies in the fact that we miss the minimum which corresponds to positioning one of the producers in the low permeability region. Hence we add in the 16 prediction points to refit the surface. Now, using the 47 points to loft the surface, we get a much better approximation. Using

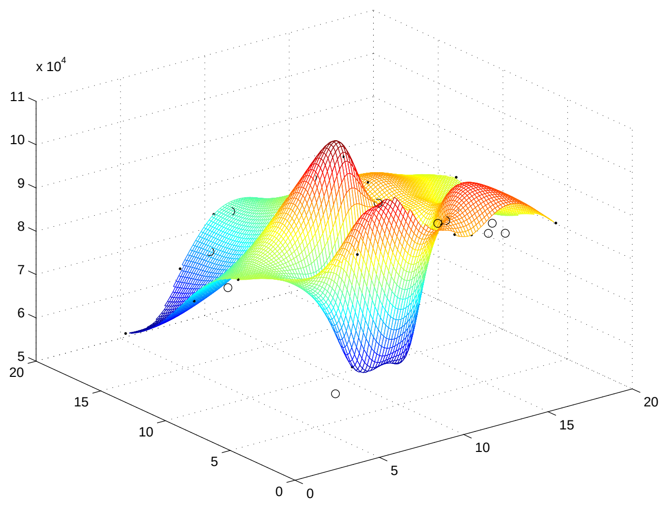


Figure 14: The 16 points we are trying to predict are are the blue circles while the 31 black points are the knots of the spline.

Newton's method, we find a local max at $(8.9892, 8.5549)$ which predicts $106,460$ barrels of oil. Translating this point on the the simulation grid puts the first producer at the point $(9,\ 14)$ and the second at $(9,\ 7)$ and yields $99,999$ barrels of oil. The surface suggests a higher value around the point $(8,\ 3)$. Newton's method does not converge because this is not a critical point of the surface. At $(8,\ 3)$, we predict 127,500 barrels of oil. The simulator only produces $99,985$ barrels of oil for the corresponding positions $(8,\ 17)$ and $(3,1)$. However this is much closer to the regression model's prediction. Fig. 15 shows these results.

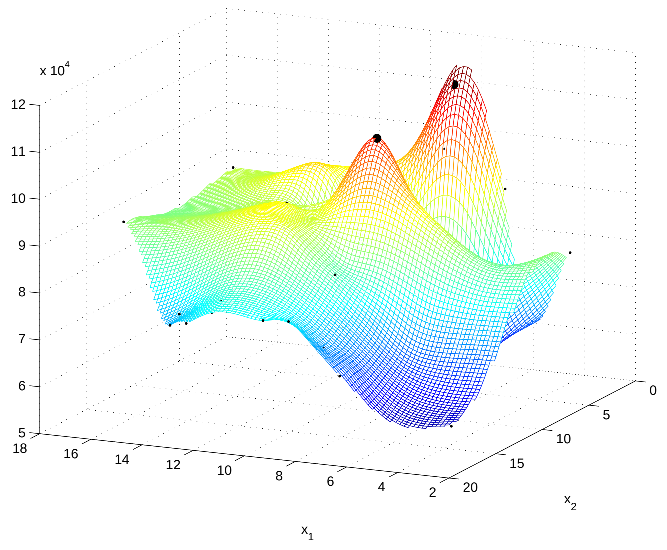


Figure 15: The surface become more dynamic using 47 points to plot instead of only 31. Note that the blue areas correspond to when one well is close to a fault. The local max in the middle of the surface marked with a large black dot is our optimation result. The large black dot near the boundary of the surface denotes our test point.

3.2.4 History Matching

For history matching we will follow the same method of using minimal energy splines to compute the surface. We will only be changing the interpretation of the surface and looking for the minimum error instead of the maximum profit. Again we will use Newton's method to find the minimum error. For our method to reduce to two variables, one end of each channel is fixed and we search for the end location that minimizes the error-norm. In Fig. 16 we find the minimum of the surface to be at (18.4941, 12.9420). This point does not match the one used to create the simulation but it creates a similar permeability field. In Fig. 17 we compare the actual permeability fieldd with permeability field generated by our guess.

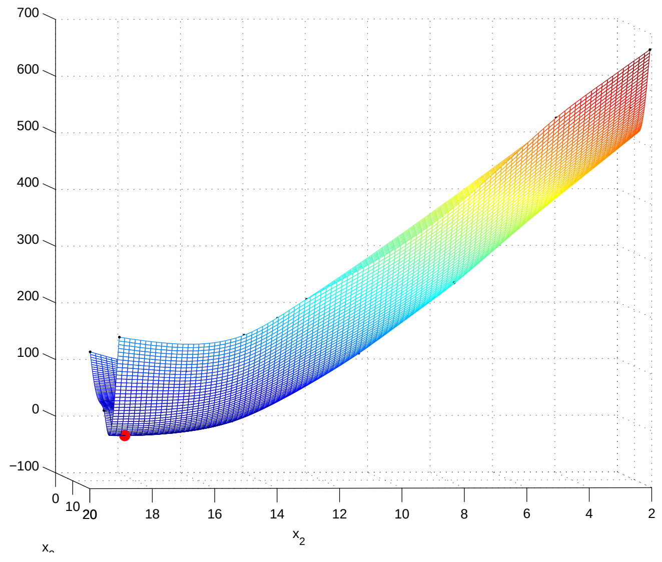


Figure 16: In the above figure we plot the error with respect to α_1 and α_2. The red dot depicts the minimum error.

3.3 Model Analysis using Neural Networks

The last subsection describes our team's efforts to use neural networks as a tool to build response surfaces for reservoir models. Given the time constraints, we didn't dive into the theory but rather performed some experiments with MATLAB's neural network toolbox.

3.3.1 Neural Networks

Neural networks (NN) were investigated first in the mid 1940s when Warren McCulloch and Walter Pitts studied the mathematical (and algorithmic) aspects of nervous activity. NN were re-discovered in the mid 1980s and introduced to a broader audience when DE Rumelhart and JL McClelland wrote a series of books about the subject. Financial institutions tried to use NN to forecast stock prices and currency exchange rates, engineers were interested in the aspects of developing control circuits for intelligent automata, and computer scientists and applied mathematicians studied the

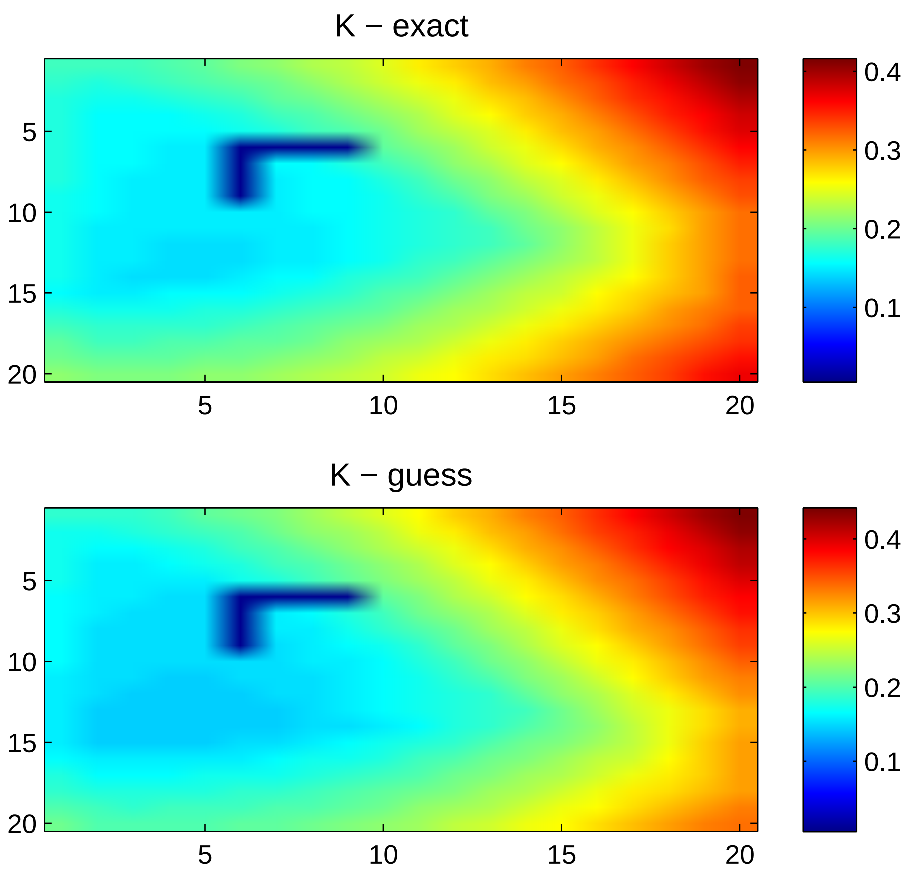


Figure 17: The top plot shows the original permeability field and the bottom plot is the estimated permeability from the spline fit.

implementation and optimization of different NN realizations. Ater a period in which interest declined, NN are again gaining in popularity because of their potential use in optimization.

What are Neural Networks?

Neural networks are patterns of interconnected cells modeled - or inspired - by the way the nervous system operates. However, compared to the complexity of the nervous system of even the smallest animal, artificial NN are trivial in structure. The basic concepts are:

- Cells (neurons) are in a state of activation or excitement
- Cells are connected with other cells
- Connections among cells propagate information in the form of signals
- Propagated signals have different amplitude, shape, duration and frequency
- Connection paths btween cells can weaken and/or delay information
- Connection weights represent knowledge that was learned before

Of course, if we wish to construct even the simplest model of how cells interact in the nervous system of an animal, we have to assume that the total input into a given neuron is a continuous function, representing a time-integrated series of nerve impulses arriving asynchronously with different frequency, strength, and duration over thousands of incoming synaptic connections. The strong interconnection between neurons introduces feedback loops and a level of implicitness that makes it extremely hard to develop a mathematical model for it. Therefore, in practical NN applications, the input to a neuron will usually be treated as a weighted sum of N input levels (derived from feeding, connected neurons), the connections will be directed such that there are no or only a few feedback loops, the activation function will be a simple shaped statistical or natural transfer function, and the system steps from one discete state to the next.

In such a network, the overall activation state of the neurons determines the state of the system, while the connection weights represent the memory of the system. Signal levels presented at one end of the feed-forward network are processed/propagated forward toward the other end of the network based on transfer functions and given connection weights. *Modifying the connection weights so that the system answers a given stimulus with an expected response* is the crux of working with NN. Much time is spend to develop *learning algorithms* that define how those weights have to be changed in order to achieve the required overall transfer function of the network.

One of the simplest feed forward networks is the perceptron, shown in the figure below:

Transfer Functions and Signal Propagation

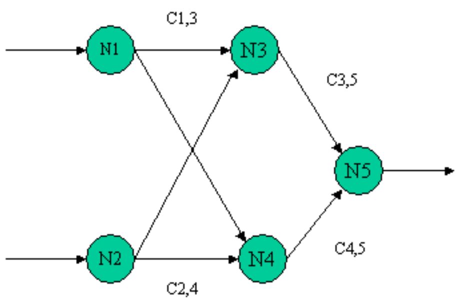


Figure 18: Simple 2-layer Perceptron

Let $C_{j,i}$ be the set of all connections leading toward neuron i. For instance, $C_{j,5}$ is the set of all connections leading toward neuron 5. The set $C_{j,5}$ contains two elements, $c_{3,5}$ and $c_{4,5}$. We can define the total weighted input into node i as:

$$y_i = \sum_j [u_j \cdot w_{j,i}] \tag{2}$$

where $w_{j,i}$ is the *connection weight* of connection $c_{j,i}$ and

$$u_j = f(y_j) \tag{3}$$

is the activation state of neuron j. The *transfer function* that calculates the activation state of a neuron based on its input can be a step-function or some linear or nonlinear combination of the weighted inputs. Sometimes, different layers within an NN use different transfer functions. Typical transfer functions are:

$x < \beta \rightarrow u(x) = 0$ else $u(x) = 1$	Step Functions
$u(x) = \alpha \cdot x$	Linear Functions
$\phi_s(i) = exp(\frac{\|s-c_i\|^2}{2\sigma_i^2})$	Radial Basis Functions
$\frac{1}{1+exp(-\beta \cdot x)}$	Logistic Functions

The choice of transfer function depends, of course, on the purpose of the network, its desired output signal shape, and its learning strategy. If the network is used to emulate a logic circuit were output signals are either 0 or 1, then at least the neurons in the output layer will have to use a step function. On the other hand, if the learning algorithm used is gradient based, then at least some neurons inside the network should use a transfer function with a non-zero first derivative.

Learning Algorithms

Different learning strategies and algorithms have been developed for NN. Some of them are listed below:

Hebbian Learning
Error Backpropagation (Local Gradient Learning)
Competitive Learning
Adaptive Resonance Theory

The usefulness of gradient based algorithms is intuitively clear and we will use one as example here. Assume we have a network such as the one shown in Fig. 18 and we want to train it to show a certain response to a given input signal. Let the desired output be $\vec{V}$ and the actual output be the activation of neurons in the last layer (#3) of the perceptron. We can define the system error to be:

$$E_S = \frac{1}{2} \cdot \sum_{k \in L_3} (V_{l(k)} - u_k)^2$$

where $l(k)$ is some mapping from the indices of the output layer neurons to the desired output vector. In our example: $E_S = \frac{1}{2} \cdot (V_1 - u_5)^2$. The derivative of the system error with respect to a given neuron j in the output layer is:

$$\frac{1}{2} \cdot \frac{\partial \left(\sum_{k \in L_3} (V_{l(k)} - u_k)^2\right)}{\partial u_j} = \frac{1}{2} \cdot \frac{\partial (V_{l(j)} - u_j)^2}{\partial u_j} = \left(V_{l(j)} - u_j\right)$$

We will call this quantity the error signal δ_j for node j. For hidden neurons and neurons in the input layer we define the error signal to be a weighted sum of error signals from the feeded neurons, times the derivative of the local input:

$$\delta_j = \left(V_{l(j)} - u_j\right) \quad \textit{Output Layer} \tag{4}$$

$$\delta_j = -\sum_{c \in C_{ji}} \delta_i \cdot w_{ji} \cdot f^{'}(y_j) \quad \textit{Hidden Layers} \tag{5}$$

Weights are then modified according to the local error gradient:

$$\Delta w_{ji} = -\delta_i \cdot y_j \tag{6}$$

Some Remarks

Since the error of hidden and input units depends on the error of neurons in front of them, this learning scheme is called Error Back Propagation. In practice, one has to deal with certain additional details to really make the theory work. For instance, if the task of the network includes mapping the zero input vector to a nonzero output vector, then - in most cases - some bias has to be applied to the inputs to allow the neurons to stay active. Also, the introduced learning scheme might be too slow and one might accelerate the weight modification by taking into account previous weight changes. Input variables and learning vectors have to be scaled to fall into the proper domain and range of the applied transfer functions. Hints and details can be found in the rich literature dealing with the subject.

3.3.2 Experimental Setup

For our experiments we used MATLABś NN toolbox. We chose a simple, 2-layer network using a radial basis function in the first layer and a linear transfer function in the second layer. We did not have the time to explore different network structures, learning algorithms, and parameter settings. Our experiments are a first step in testing the applicability of NN in a reservoir simulation environment. Two experiments were performed:

1. Production Prediction Experiment
2. Well Placement Experiment

The third experiment, which tried to predict the permeability field, was not performed.

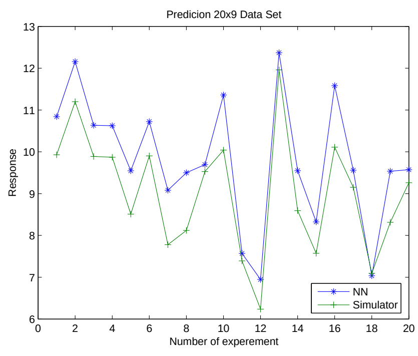


Figure 19: Neural Network Prediction of 3-Domain Model

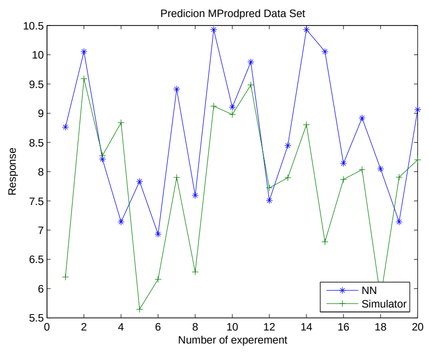


Figure 20: Neural Network Well-Placement Experiment

3.3.3 Results

The neural network was able to predict a model's trend with good accuracy, however it failed to give an exact quantitative prediction of the expected result. Our assumption is, that the network needs more training points to increase it's accuracy. In Fig. 19 we show the prediction of the NN for the 3-domain model a training data-set with 20 entries. Remember, that this model had 3 tuneable parameters in each of the 3 subdomains. So, we have only 20 datapoints to predict a model with ≤ 9 degrees of freedom. The NN is able to establish a releationship between these parameters and the total oil production, however it fails to give an exact guess for the produced volume.

The same observation can be made for the well-placement experiment. The NN is able to predict the trend for different well locations pretty well but fails to answer how large the production in these locations really is. Fig. 20 shows the results.

4 Conclusions

We have tested three approaches to build surrogate models for simple oil-reservoirs. Models based on regression analysis and splines were able to capture important parameter dependencies and predict total production. The neural network model was able to accurately predict parameter dependencies and model trends but failed to make acceptable quantitative predictions. The tested models needed a relatively large batch of data samples. This can become a problem in more complicated, real life scenarios where simulation runs are often very costly and time-consuming. Overall, the results are very encouraging since the applied mathematical techniques are straightforward and leave room for optimization.

References

- William H. Greene, Econometric Analysis, Fifth Edition, Pearson Education.
- André I. Khuri, John A. Cornell, Response Surfaces Designs and Analysis, Second Edition, Marcel Dekker, Inc.
- G. Awanou, M. J. Lai, P. Wenston, The multivariate spline method for numerical solution of partial differential equations, Wavelets and Splines, 2006, 24-74.
- R. Eckmiller, Christoph v.d. Malsburg (Eds.), Neural Computers
- Stephen Grossberg (Editor), Neural Networks and Natural Intelligence
- J. L. McClelland, D. E. Rumelhart, Explorations in Parallel Distributed Processing