

SPARSE PROBABILISTIC COLLOCATION FOR UNCERTAINTY QUANTIFICATION IN RESERVOIR ENGINEERING

J. H. Mendes¹, R. B. Willmersdorf¹

¹ Department of Mechanical Engineering, Center of Technology and Geosciences at Federal University of Pernambuco (julio.hoffmann@gmail.com, ramiro@willmersdorf.net)

Abstract. *This paper presents an application of uncertainty quantification to numerical reservoir simulation using the Sparse Probabilistic Collocation Method (SPCM). Reservoir simulation is used in several phases of the development and exploitation of a field, from the initial planning of the production strategy to sophisticated automated control schemes that schedule the operation of well controls on a daily basis in smart fields. It basically consists of solving numerically the complex nonlinear partial differential equations (PDEs) that models the fluid flow in porous media. The petrophysical properties of the rock matrix determine the coefficients in the PDEs and associated algebraic system of equations. Due to technological and economic constraints, the available data to determine these properties is scarce and subject to human interpretation. This problem becomes even more important for offshore fields, where wells are kilometers apart, the reservoirs several kilometers underground, there are very few wells and there is very little or no production history. The petrophysical properties are therefore very uncertain, and can be described only in a probabilistic manner. The simulation can no longer be considered deterministic, since uncertain inputs leads to uncertain results. Uncertainty propagation techniques become necessary tools for the robust and reliable application of numerical reservoir simulation. In the probabilistic collocation method, statistics of the uncertain output are computed directly through numerical integration, based on efficient quadrature rules like Gauss and Clenshaw-Curtis. However, this method is not suitable for dealing with high-dimensional models, because it suffers from the “curse of dimensionality”. Sparse grid integration techniques can be used with the probabilistic collocation method to alleviate this problem, creating the sparse probabilistic collocation method. We present our implementation of the SPCM and apply it to estimate the statistics of uncertain variables such as the cumulative oil production and water breakthrough date, using a simple but realistic reservoir model. Comparisons of the efficiency of this technique against classical methods such as Monte Carlo are shown, as well as a discussion on the necessary computational resources for this kind of analysis and it’s practical use.*

Keywords: *Polynomial chaos, Stochastic collocation, Sparse grids, Reservoir engineering.*

1. INTRODUCTION

Uncertainty quantification is used in almost all oil industry activities. From data acquisition to economic evaluations, companies are applying advanced techniques to better characterize the field and its production life. The reservoir engineering team is particularly interested in quantifying uncertainty during the fluid flow simulation [7], where the lack of knowledge about the petrophysical properties and geological formations lead to unsafe predictions. One could argue that, sometimes scarce data isn't a real issue because 3D seismic is available even in the presence of geological signal attenuators. Unfortunately, the uncertainty is still present; geological interpretation is one of the most notorious sources of it.

Uncertainty due to lack of knowledge is called epistemic uncertainty and is being studied with alternative theories other than probability theory. Eldred et al purposed an encapsulation method trying to avoid strong structures on the data [9]. In this paper we adopted the probabilistic approach [5] though, because it's well established in several application areas and is consistent with the main geostatistical approaches used for reservoir modeling.

Most current reservoir simulators are essentially sequential solvers, it's not feasible to treat uncertainty without some form of parallel processing because each run could take hours to complete. The probabilistic collocation method [10] uses a decoupled formulation such that is trivial distribute the runs among many computational nodes. In [2], the authors made very clear two important features of stochastic collocation: high convergence rate from polynomial chaos expansions and easy implementation as found in sampling methods.

In the next section, we give a brief timeline of the recent work on probabilistic collocation, how it was developed chronologically and the reasons for such. Next, we review the method, as simply as possible, to clarify what we understand by probabilistic collocation. The general Smolyak combination technique is then explained, showing that is easy to achieve parallelism by *task farming*. Finally, we apply the SPCM to a reservoir model to better illustrate the workflow of adapting a deterministic solver to uncertainty quantification.

2. BACKGROUND

Several researchers on stochastic analysis have been developing efficient alternatives for the widely employed sampling methods used in uncertainty quantification (UQ). The need to decrease the number of simulator runs when estimating statistics of interest in problems where each run is computationally expensive has led to the study of high-order polynomial expansions.

This first attempt was characterized by an intense study of the polynomial chaos expansion (PC) itself and error estimation [6]. The PC strategy was invaluable to derive convergence rates, giving a solid mathematical basis for the subsequent methods here discussed.

Concurrently, new polynomial families were included to treat non Gaussian probability distributions. Xiu and Karniadakis [12] presented a map between probability density functions and orthogonal polynomial families which is known as the Wiener-Askey scheme. Within this scheme, the developer could choose the optimal polynomial family for obtain the desired exponential convergence rate. This technique became known as Generalized Polynomial Chaos (gPC) [13].

The main reason to develop alternative methods was the fact that, in the PC approach using stochastic Galerkin projections, the underlying system of equations becomes coupled. The implementation could be very difficult, especially in cases where non-linearities and/or coupled multiphysics are present. To overcome that inconvenience, the target deterministic solver was modeled as a *black box* and the collocation techniques emerged.

Collocation based methods were very important for application of UQ to problems with complex physics. The complexity of the target solver could be ignored and the only requirement was a smooth response of interest over the parametric domain. At this point, each probability distribution induced an efficient quadrature rule associated to a polynomial basis given by the Wiener-Askey scheme. If all random variables were independent in the non-deterministic model, nothing would be more natural than employing multivariate quadrature rules to estimate statistics.

3. PROBABILISTIC COLLOCATION METHOD

Let $\xi(\omega) = [\xi_1(\omega), \xi_2(\omega), \dots, \xi_n(\omega)]$ be a random vector mapping $\Omega \mapsto \Gamma \subset \mathbb{R}^n$ and $\rho = \prod_{i=1}^n \rho_i$ it's joint probability density function (pdf). For any well-behaved function $f : \Gamma \subset \mathbb{R}^n \mapsto \mathbb{R}$ of these random variables, the first two statistical moments are given by:

$$\mathbb{E}[Y = f(\xi)] = \int_{\Gamma} f(\xi_1, \xi_2, \dots, \xi_n) \prod_{i=1}^n \rho_i d\Gamma \quad (1)$$

$$\mathbb{V}[Y = f(\xi)] = \int_{\Gamma} [f(\xi_1, \xi_2, \dots, \xi_n) - \mathbb{E}[Y]]^2 \prod_{i=1}^n \rho_i d\Gamma \quad (2)$$

By exploiting independence, these integrals can be decoupled for each density function and a multivariate quadrature is direct constructed from univariate rules.

For a general one dimensional integral of the form $\int_a^b w(x)f(x)dx$, where $w(x) \geq 0$ over (a, b) is a measurable weight function and $a, b \in [-\infty, +\infty]$ are symbols on the extended line, there are efficient quadrature rules. And more important, the most common used pdfs seen as weight functions induces Gaussian quadratures as the optimal choice for numerical evaluation:

Table 1. Linkage between probability distributions and quadrature rules

Distribution	PDF	Weight	Support Range	Quadrature Rule
Uniform(a, b)	$\frac{1}{b-a}$	1	$[-1, 1]$	Gauss-Legendre $P_n(x)$
Normal(μ, σ^2)	$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}}$	e^{-x^2}	$[-\infty, \infty]$	Gauss-Hermite $He_n(x)$
Log-normal(μ, σ^2)	$\frac{1}{x\sqrt{2\pi}\sigma} e^{-\frac{1}{2} \frac{(\ln x - \mu)^2}{\sigma^2}}$	e^{-x^2}	$(0, \infty]$	Gauss-Hermite $He_n(x)$
Exponential(λ)	$\lambda e^{-\lambda x}$	e^{-x}	$[0, \infty]$	Gauss-Laguerre $L_n(x)$
Gamma(α)	$\frac{x^\alpha e^{-x}}{\Gamma(\alpha+1)}$	$x^\alpha e^{-x}$	$[0, \infty]$	Gauss-Laguerre $L_n^{(\alpha)}(x)$

Thus, after normalization of the support range, the appropriate univariate rule is chosen by table lookup and all the work is done by gathering the contributions of individual dimensions: the collocation grid is generated by Cartesian product and the multi-dimensional weights by the product of weights for each dimension.

As mentioned earlier, the probabilistic collocation method is really simple to implement, given the code for univariate rules. One problem that arise from this approach is related to the *curse of dimensionality*: the number of collocation points grows exponentially with the number of dimensions in the stochastic model. This implies that the probabilistic collocation method described so far is not adequate for high-dimensions and the parametric space must be small for the problem to be tractable.

A combination technique for multivariate quadrature formulas was developed a part of the context of uncertainty quantification by Smolyak [11]. Gerstner and Griebel [8] studied further the Smolyak algorithm and it's variants pushing researchers to use it in many different areas.

4. GENERALIZED SMOLYAK COMBINATION

To explain the Smolyak technique we need to introduce some auxiliary constructs. Let's begin by indexing 1D quadrature rules of the same type with increasing order:

$$\{\mathcal{U}^1(f), \mathcal{U}^2(f), \dots, \mathcal{U}^i(f), \dots\} \quad (3)$$

Here, the position in the list is referred as the level. Each level has an order associated with, depending on the rule type or some another external knowledge:

$$\mathcal{U}^i(f)(\xi) = \sum_{j=1}^{m_i} w_j^i f(\xi_j^i) \quad (4)$$

The order m_i can take different forms such as linear $m_i = i$; powers of two $m_i = 2^i - 1$; etc. Quadrature nesting is also related to that functional form or *order growth*, and will be recap later.

4.1. Isotropic interpolant

Now, for clarity, assume all n dimensions are being approximated with the same quadrature rule above. Probabilistic collocation, as described earlier in it's purest version, is to set n orders $m_{i_1}, m_{i_2}, \dots, m_{i_n}$ and take the tensorial product:

$$\mathcal{Q}(f) = (\mathcal{U}^{i_1} \otimes \dots \otimes \mathcal{U}^{i_n})(f) = \sum_{j_1, j_2, \dots, j_n=1}^{m_{i_1}, m_{i_2}, \dots, m_{i_n}} (w_{j_1}^{i_1} \otimes \dots \otimes w_{j_n}^{i_n}) f(\xi_{j_1}^{i_1}, \dots, \xi_{j_n}^{i_n}) \quad (5)$$

Note that i_1, i_2, \dots, i_n are fixed levels for which the chosen orders match, and the obtained full grid has $\prod_{j=1}^n m_{i_j}$ collocation points. Remembering Lagrange interpolation, the following Pascal's triangle illustrates the inefficiency in two dimensions:

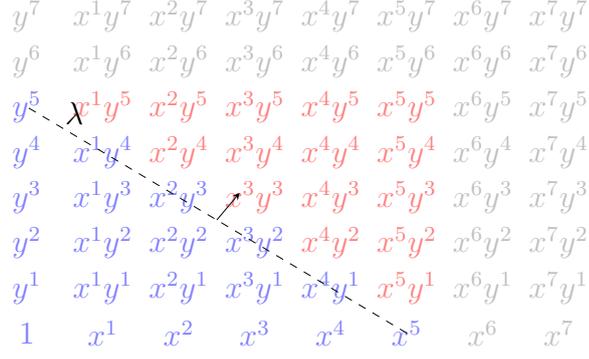


Figure 1. Pascal's triangle arrangement showing the full product inefficiency. Desired monomials in blue, undesired monomials in red.

Only “half” of the monomials in the full product contributes for the desired precision. The incomplete diagonals in the above scheme inserts errors of the same order and the goal is to eliminate those extra monomials drawn in red. Formally, define the sets:

$$X(w, n) := \left\{ \mathbf{i} \in \mathbb{N}^n, \mathbf{i} \geq \mathbf{1} : \sum_{k=1}^n (i_k - 1) \leq w \right\}, \quad (6a)$$

$$Y(w, n) := \left\{ \mathbf{i} \in \mathbb{N}^n, \mathbf{i} \geq \mathbf{1} : w - n + 1 \leq \sum_{k=1}^n (i_k - 1) \leq w \right\} \quad (6b)$$

where $w \in \mathbb{N} = 1, 2, \dots$ is the maximum allowed level sum. Also, define the incremental interpolant $\Delta^i := \mathcal{U}^i - \mathcal{U}^{i-1}$ by setting $\mathcal{U}^0 = 0$. The isotropic Smolyak algorithm is presented in two forms [5], with completely different implementations:

$$\mathcal{A}(w, n) = \sum_{\mathbf{i} \in X(w, n)} (\Delta^{i_1} \otimes \dots \otimes \Delta^{i_n}) \quad (7a)$$

$$= \sum_{\mathbf{i} \in Y(w, n)} (-1)^{w+n-|\mathbf{i}|} \binom{n-1}{w+n-|\mathbf{i}|} \cdot (\mathcal{U}^{i_1} \otimes \dots \otimes \mathcal{U}^{i_n}) \quad (7b)$$

The second equation (Eq. 7b) is a explicit combination of low level tensor products where $|\mathbf{i}| := i_1 + \dots + i_n$ is the sum of entries in the multi-dimensional Smolyak index. There is no additional work other than finding the indices and combining the rules. However, the task of finding indices can be challenging in high-dimensions, the cost of naive loop-based implementations is prohibitive. We propose an algorithm based on integer partitions as a viable solution, Algorithm 1.

The first equation above (Eq. 7a) has a great impact on the performance when nested quadrature rules are being applied. These rules produce nested grids, meaning that as the level is increased, many or all collocation points overlap. An efficient implementation should keep track of the lower order grid evaluations to use in posterior higher order interpolants.

Is important to note that the order growth can be more aggressive with nested rules, where the number of new evaluations is reduced compared to a non-nested scenario. For refer-

Input: (w, n) - level sum and dimensionality

Output: M - matrix whose rows are the Smolyak indices

```
// matrix size
1 for  $m \leftarrow \max(n, w + 1)$  to  $w + n$  do
2 |    $nrows \leftarrow nrows + \binom{m-1}{n-1}$ 
3 end
4  $ncols \leftarrow n$ 
5  $M \leftarrow \text{zeros}(nrows, ncols)$ 

// filling the matrix
6 for  $m \leftarrow \max(n, w + 1)$  to  $w + n$  do
7 |    $maxElm \leftarrow m - n + 1$ 
8 |    $row \leftarrow (1, 1, \dots, 1, maxElm)$ 
9 |   Assign row to  $M$ 
10 |  while  $firstElm(row) < maxElm$  do
11 |    |  $pos \leftarrow lastPos(row)$ 
12 |    | while  $row[pos] = 1$  do  $pos \leftarrow pos - 1$ 
13 |    |  $row[pos - 1] \leftarrow row[pos - 1] + 1$ 
14 |    |  $row[pos] \leftarrow row[pos] - 1$ 
15 |    | if  $pos \neq lastPos(row)$  then
16 |    | |  $swap(row[pos], row[lastPos(row)])$ 
17 |    | end
18 |    | Assign row to  $M$ 
19 |  end
20 end
21 return  $M$ 
```

Algorithm 1: Compute Smolyak indices

ence: Clenshaw-Curtis and Gauss-Patterson are nested, Gauss-Legendre and Gauss-Hermite are weakly nested, Gauss-Laguerre isn't nested, and in general, Gauss rules are not nested [1].

4.2. Anisotropic interpolant

In many cases, the model is known beforehand to be anisotropic, making the use of the same maximum order for all random variables inadequate. The isotropic Smolyak algorithm can be adapted to fit the more critical dimensions first, through a simple geometric idea.

By looking at the Pascal's triangle (Figure 1), the λ -line represents the frontier in traversing the monomials with the same total degree uniformly. The line slope can be manipulated to a value other than 45° producing anisotropic formulas.

The predicates in the description of the auxiliary sets are modified to accommodate an anisotropy vector $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{R}_+^n$:

$$X_\alpha(w, n) := \left\{ \mathbf{i} \in \mathbb{N}^n, \mathbf{i} \geq \mathbf{1} : \sum_{k=1}^n (i_k - 1)\alpha_k \leq w\underline{\alpha} \right\}, \quad (8a)$$

$$Y_\alpha(w, n) := X_\alpha(w, n) \setminus X_\alpha(w - \frac{|\alpha|}{\underline{\alpha}}, n) \quad (8b)$$

$$= \left\{ \mathbf{i} \in \mathbb{N}^n, \mathbf{i} \geq \mathbf{1} : w\underline{\alpha} - |\alpha| < \sum_{k=1}^n (i_k - 1)\alpha_k \leq w\underline{\alpha} \right\} \quad (8c)$$

where $|\alpha| := \alpha_1 + \dots + \alpha_n$ and $\underline{\alpha} := \min(\alpha_1, \dots, \alpha_n)$. The anisotropic Smolyak formula, in it's combinatorial representation, is then written as [3]:

$$\mathcal{A}(w, n, \alpha) := \sum_{\mathbf{i} \in Y_\alpha(w, n)} c_\alpha(\mathbf{i}) \cdot (\mathcal{U}^{i_1} \otimes \dots \otimes \mathcal{U}^{i_n}) \quad (9)$$

with the $c_\alpha(\mathbf{i})$ coefficients given by:

$$c_\alpha(\mathbf{i}) := \sum_{\substack{\mathbf{j} \in \{0,1\}^n \\ \mathbf{i} + \mathbf{j} \in X_\alpha(w, n)}} (-1)^{|\mathbf{j}|} \quad (10)$$

That means, for each index $\mathbf{i} \in Y_\alpha(w, n)$, find all perturbation vectors \mathbf{j} whose entries are 0 or 1, such that $\mathbf{i} + \mathbf{j}$ is still on the sparse grid. The coefficient $c_\alpha(\mathbf{i})$ is a sum over all it's acceptable perturbations where again, $|\mathbf{j}| := j_1 + \dots + j_n$.

By restricting $\alpha \in \mathbb{Z}_+^n$ in the library design, Algorithm 1 can be easily adapted for the anisotropic interpolant. It was not shown due to technical constraints on the article size. For the general case with real entries, a different implementation is required [1].

Yet more general formulas, like "corner" grids, were constructed for sparse interpolants, all sharing similar ideas explained in this section, built on top characteristic sets [8].

5. APPLICATION TO A RESERVOIR MODEL

Having explained the SPCM (PCM + Generalized Smolyak algorithm), we're now able to apply the method to a simple, yet realistic, reservoir model. Consider the following problem description:

5.1. Deterministic characterization

$44 \times 24 \times 5$ blocks, each with $200 \times 200 \text{ m}^2$ basis area, adding up to 42.24 km^2 field extension, are located approximately between $2958\text{--}3425 \text{ m}$ depth, their thickness varies block to block according to a map assembled from geological processing.

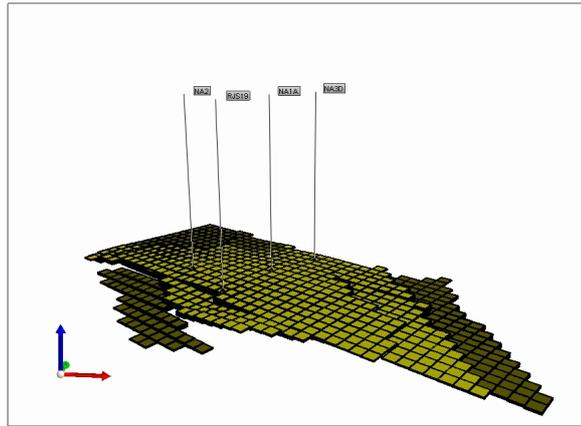


Figure 2. Reservoir numerical grid

Figure 2 shows the reservoir structure. The anticline formation is represented with gradient colors translating macroscopically the porosity of each layer. Darker regions at the bottom present lower porosity as in the majority of reservoir models. This phenomena is often referred in geology as the *stratigraphic deposition principle*, in which the more recent layers act compressing the older ones below it. Table 2 contains the important petrophysical properties for this case study: effective porosity and absolute permeabilities. Hereafter, the top layer is labeled “Layer 1”.

Table 2. Petrophysical properties per layer and from top to bottom

Layer	ϕ_{ef}	κ_x	κ_y	κ_z
1	0.15	550mD	550mD	220mD
2	0.15	—	—	—
3	0.15	—	—	—
4	0.14	—	—	—
5	0.13	—	—	—

The effective compressibility is $1 \times 10^{-4} (\text{kgf}/\text{cm}^2)^{-1}$ at $1.033 \text{ kgf}/\text{cm}^2$ reference pressure, the pore volume is 165588 m^3 and the volumes of the fluids in place are given in

Table 3. There is no gas in place, all produced gas is the result of phase transformation along the isothermal ($88^{\circ}C$) elevation. Hence, it's a black oil reservoir with no gas cover, but with a considerably large subjacent aquifer that will guide the production, due to the water influx mechanism, see Figure 3. A gas-liquid ratio of $RGL \approx 0.11m^3std/m^3std$, as well as a minimum reservoir pressure of $P_{min} \approx 316kgf/cm^2$, higher than the bubble pressure $P_b = 211kgf/cm^2$, both ratify the previous affirmative.

Table 3. Fluid volumes

Fluid	Volume(m^3)
Oil STD	60830
Total gas	7024
Free gas	—
Water	79808
Oil in place	85413
Gas in place	—
Water in place	80175

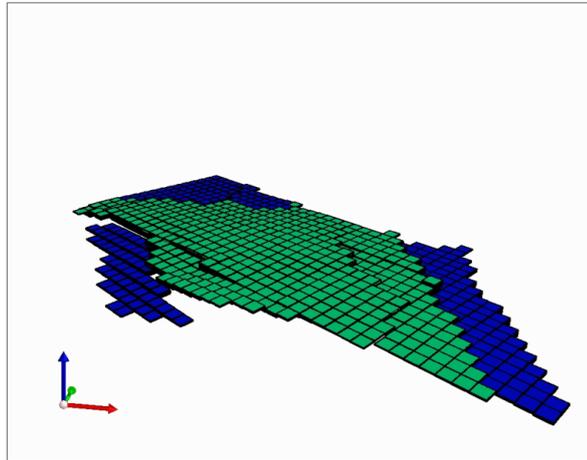
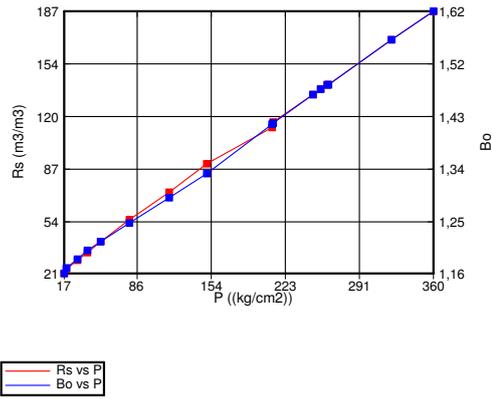


Figure 3. Reservoir subjacent aquifer. Oil-water contact 3105m deep.

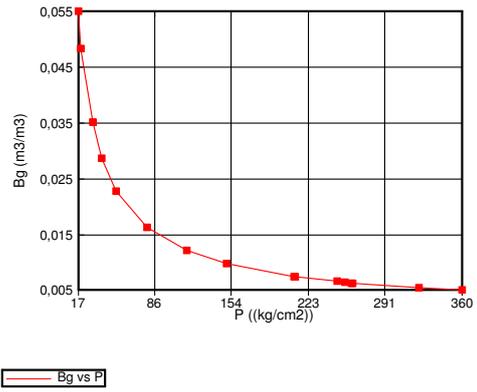
The oil and water average saturations can be calculated using the volumes in Table 3 and void space:

$$\bar{S}_o = \frac{85413m^3}{165588m^3} \approx 51.6\% \quad \bar{S}_w = \frac{80175m^3}{165588m^3} \approx 48.4\% \quad (11)$$

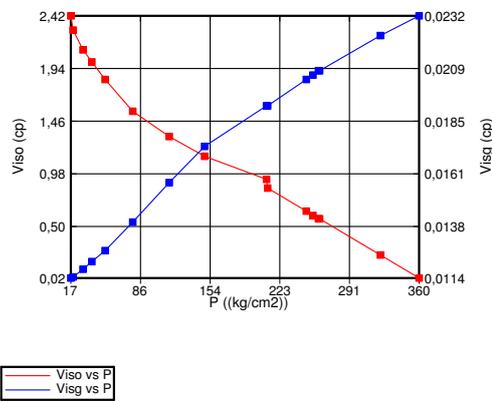
As noted, there is no natural gas, $\bar{S}_o + \bar{S}_w = 100\%$. The oil and gas densities are $\rho_o = 890kg/m^3$ and $\rho_g = 0.8582kg/m^3$, respectively. The oil compressibility beyond the bubble point is $1.2 \times 10^{-4}(kgf/cm^2)^{-1}$ and Figure 4 groups the volume formation factor, solubility ratio, viscosity, and relative permeability curves for these hydrocarbons. The water properties are listed for the reference pressure of $1.033kgf/cm^2$: compressibility, $4.6 \times 10^{-5}(kgf/cm^2)^{-1}$; viscosity, $0.4cp$; and volume formation factor, 1.02.



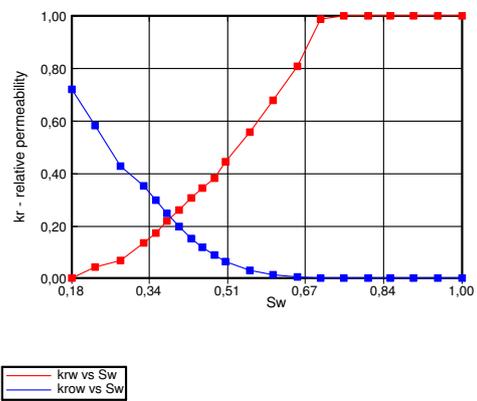
(a) Solubility ratio and oil volume formation factor



(b) Gas volume formation factor



(c) Oil and gas viscosities

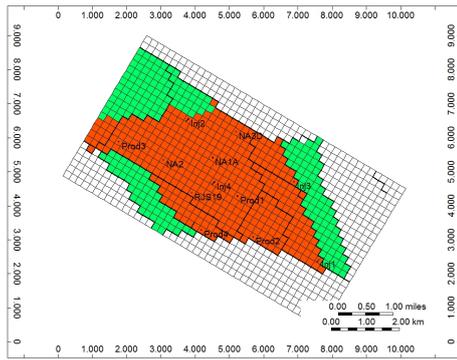


(d) Oil-water relative permeabilities

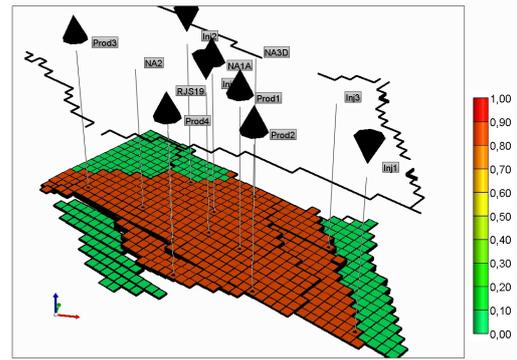
Figure 4. Volume formation factors, solubility ratio, viscosities and relative permeabilities as functions of either, pressure or water saturation

Based on the field production mechanism, the oil recovery strategy main directive is to preserve the aquifer energy. It's importance is in maintaining the exploitation stable at high-pressure levels. Thus, the producer wells locations/completions must be sufficiently far from oil-water contacts to delay production of water. Also, injector wells are placed in the oil accumulation for pushing the fluids in a radially convergent manner.

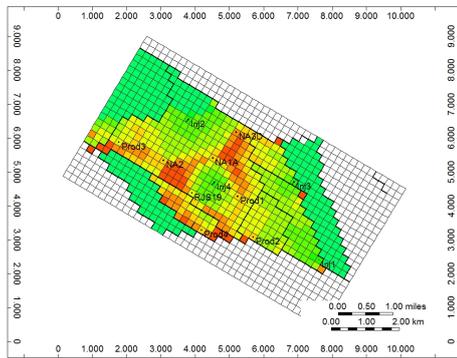
Figure 5 contains top and 3D views of all 12 existing wells. Completions vary from 3 to 5 layers, but almost all wells have full completion. They operate with two types of constraints, primary flow constraints and secondary pressure constraints. For example, $1100\text{m}^3/\text{s}$ is the maximum allowed flow for the NA1A producer, and it's minimum bottom-hole pressure is set to $190\text{kgf}/\text{cm}^2$. The distinction made between primary and secondary constraints is a technical detail of the CMGTM software [4] that specifies only primary constraints are subject to posterior modification, a commonly used feature in history matching. Besides these constraints, wells are closed in the case of backflow. Figure 6 shows the cumulative oil production for the field. All deterministic and stochastic studies were done with IMEXTM reservoir simulator software [4].



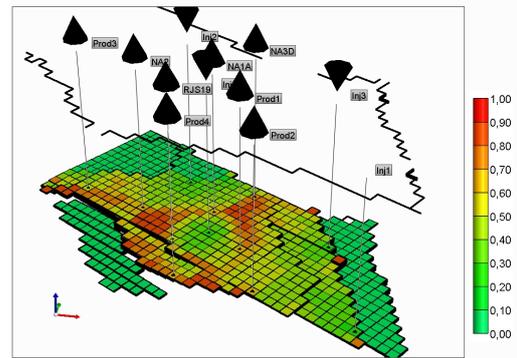
(a) Wells top view



(b) Wells 3D view



(c) Oil saturation at the end of simulation (Top view)



(d) Oil saturation at the end of simulation (3D view)

Figure 5. Oil saturation before and after simulation. 12 wells - Producers: NA1A, NA2, NA3D, RJS19, Prod[1-4]; Injectors: Inj[1-4].

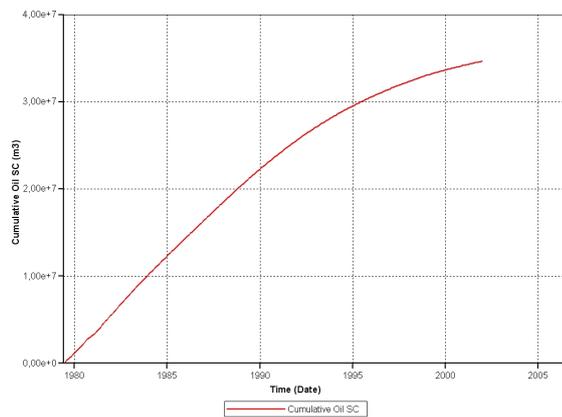


Figure 6. Cumulative oil production

5.2. Modeling uncertainty

It's crucial to define what are the most influential input variables and what probability distributions, if applicable, they should be assigned. Scientific methods for doing parametric studies and weighted selection are available, and should be used in real life scenarios. Since

this article aims to demonstrate the technique, this phase was neglected in favor to other tasks.

A common approach is to give the petrophysical properties, such as absolute permeabilities, log-normal distributions. With today's technology is nearly impossible to infer the subsurface stress state that affects those properties, even in well proximities. Bringing up the rock plug to standard conditions causes exudation side effects, i.e. depressurization can alter the physical-chemical state of the specimen, which also invalidates the laboratory analysis. And, of course, in real reservoir models, the assignment of petrophysical properties to cell blocks is done with geostatistical algorithms that are stochastic by nature. Therefore, porosities and permeabilities are good candidates for uncertainty modeling.

Small case studies were created due to time limitation, but also for showing the library flexibility. At the end, scalability issues and possible solutions are discussed. Monte Carlo simulations were run for each case to verify our implementation, and to assess its computational efficiency.

The analysis were run on a time shared 6.13 TFlops computing cluster of CENAPAD-PE. This machine has 72 (Bull R421®) computing nodes, each containing 2 Quad Core 2.66MHz Intel Xeon® 5355 processors. The machine runs Red Hat™ GNU/Linux. Due to licencing constraints, at most 100 IMEX™ could be done simultaneously.

For each case, the method's parameters were chosen in such a way the resultant grid had the same maximum one-dimensional quadrature order. As an example, it's known that for a sparse level w , the maximum order used in the isotropic sparse grid is at the quadrature level $w + 1$. If the nonlinear sparse growth is chosen ($m_1 = 1$ and $m_i = 2^{i-1} + 1$, $i > 1$), then the maximum quadrature order is $m_{max} = 2^{(w+1)-1} + 1 = 2^w + 1$. See Figure 7.

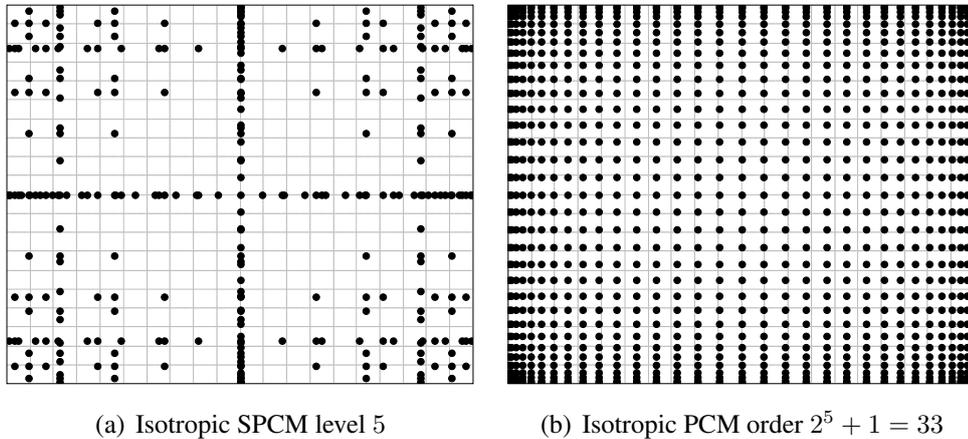


Figure 7. 2D collocation grids using Gauss rule, reproducing Fig. 1 in [5]

CASE 1. Based on Table 2, $\phi_{ef_1} \sim \mathcal{U}(0.13, 0.16)$ and $\phi_{ef_2} \sim \mathcal{U}(0.13, 0.16)$ were assigned two equally uniform-distributed random variables (R.V.). The isotropic, nonlinear, SPCM level 5 ($m_{max} = 33$) was applied, PCM and Monte Carlo were applied with 33×33 points. The response of interest was the net present value of the field computed by the simplified formula:

$$NPV = \sum_{\tau=0}^T \frac{F_{\tau}}{(1+j)^{\tau}} \quad (12)$$

where F_τ is the cash flow at time τ and j is the constant applied discount factor. The cash flow is given by $F_\tau = R_\tau - Opeex_\tau$. The oil revenue $R_\tau = P_\tau^o \cdot q_\tau^o \cdot \Delta t$ and operational expenditure $Opeex_\tau = \sum_{l=w, w_i} C_\tau^l \cdot q_\tau^{*l} \cdot \Delta t$ due to water injection and production were attributed constant prices $P_\tau^o = 30\text{U}\$/\text{bbl}$, $C_\tau^w = 5\text{U}\$/\text{bbl}$ and $C_\tau^{w_i} = 2\text{U}\$/\text{bbl}$. The minimum attractive rate of return at year was set to 0.093.

Table 4. CASE 1

	# Points	Wall Time (40 procs)	Statistics ($\mu \pm \sigma$)
SPCM	371	00:23 h	22808.3U\$ \pm 7286.7U\$
PCM	1089	01:06 h	22757.4U\$ \pm 7317.0U\$
MC	1089	01.05 h	22197.3U\$ \pm 7237.4U\$

CASE 2. $\phi_{ef_1} = \phi_{ef_2} = \phi \sim \mathcal{U}(0.13, 1.16)$ as the first R.V. and $\kappa_{x_1} = \kappa_{y_1} = \kappa_{x_2} = \kappa_{y_2} = \kappa_{\text{hor}} \sim \log \mathcal{N}(\mu, \sigma^2)$ such that $\mathbb{E}[\kappa_{\text{hor}}] = 550mD$ and $\mathbb{V}[\kappa_{\text{hor}}] = (10mD)^2$ as the second.* The explicit formulas for μ and σ^2 are given in Eq. 13. It was decided the permeability had more impact on the net present value by setting the anisotropy vector $\alpha = (2, 1)$. The tensor products used by the Smolyak algorithm (Eq. 14) guided the maximum order for PCM and Monte Carlo, $m_{\text{max}} = 2^{(6-1)} + 1 = 33$, the orders were then chosen $22 \times 33 = 726$ grid points.

$$\sigma^2 = \log(1 + V/E^2) \quad (13a)$$

$$\mu = \log(E) - \sigma^2/2 \quad (13b)$$

$$\mathcal{A}(5, 2, (2, 1)) = -\mathcal{U}^1 \otimes \mathcal{U}^4 - \mathcal{U}^2 \otimes \mathcal{U}^2 + \mathcal{U}^1 \otimes \mathcal{U}^6 + \mathcal{U}^2 \otimes \mathcal{U}^4 + \mathcal{U}^3 \otimes \mathcal{U}^2 \quad (14)$$

Table 5. CASE 2

	# Points	Wall Time (40 procs)	Statistics ($\mu \pm \sigma$)
SPCM	93	00:07 h	22664.7U\$ \pm 10417.6U\$
PCM	726	00:45 h	22664.1U\$ \pm 10417.1U\$
MC	726	00:44 h	22906.1U\$ \pm 10316.7U\$

CASE 3. The previous case with the additional R.V. $\kappa_{z_1} = \kappa_{z_2} = \kappa_{\text{vert}} \sim \log \mathcal{N}(\mu, \sigma^2)$ such that $\mathbb{E}[\kappa_{\text{vert}}] = 220mD$ and $\mathbb{V}[\kappa_{\text{vert}}] = (5mD)^2$. SPCM with $w = 5$, $\alpha = (2, 1, 1)$, PCM and Monte Carlo containing $22 \times 33 \times 33$ points ($m_{\text{max}} = 33$)[†]. The response of interest is the cumulative oil production.

*In reality, porosity and permeability are correlated.

[†]PCM with 33 points is generally overmeasure.

Table 6. CASE 3

	# Points	Wall Time (50 procs)	Statistics ($\mu \pm \sigma$)
SPCM	695	00:13 h	$34279.4m^3 \pm 638.2m^3$
PCM	23958	20:01 h	$34183.2m^3 \pm 1562.8m^3$
MC	23958	1.1 day	$34174.8m^3 \pm 1944.4m^3$

6. CONCLUSIONS

The benefit of applying SPCM becomes more evident as the dimensionality grows. The accuracy is not compromised and the computational effort is significantly reduced. Despite that fact, it's not scalable to very high dimensions. For higher dimensions ($\gtrsim 50$), even a 2-point integration rule is prohibitive, and the Monte Carlo Method becomes attractive again.

Adaptive sparse grids on top the Smolyak algorithm here discussed are being developed by the UQ research community. The incremental form presented in Eq. 7a allows the definition of *hierarchical surpluses* which serve as error estimators for an automatic refinement procedure [2]. This approach answers the question: What SPCM level to use? Because the grid is constructed with the minimum level and refined on regions with poor approximation, the user doesn't need to specify a sparse level. Scalability issues are not improved though.

State of the art techniques for uncertainty quantification are being developed and implemented within DAKOTA: An Open Source Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification and Sensitivity Analysis. Although the work presented here was done with a home grown UQ library, the authors plan to use and contribute with DAKOTA in the future.

Acknowledgements

The authors thank John Burkardt for making available many quadrature rule libraries under the LGPL license, CENAPAD-PE for providing the computational resources, and Dr. Ana Paula Costa for providing the basis reservoir model and advice on the deterministic simulation.

References

- [1] Brian M. Adams, Keith R. Dalbey, Michael S. Eldred, David M. Gay, Laura P. Swiler, William J. Bohnhoff, John P. Eddy, Karen Haskell, and Patricia D. Hough. *DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis*. Sandia National Laboratories, 5.0+ edition.
- [2] Nitin Agarwal and N. R. Aluru. A domain adaptive stochastic collocation approach for analysis of mems under uncertainties. *Journal of Computational Physics*, 2009.

- [3] John Burkardt. Advanced numerical methods for computing statistical quantities of interest from solutions of stochastic partial differential equations. Technical report, Interdisciplinary Center for Applied Mathematics/Information Technology Department - Virginia Tech, August 2010.
- [4] CMG. *IMEX Advanced Oil/Gas Reservoir Simulator*. Computer Modelling Group Ltd., Office #150, 3553 - 31 Street N. W. Calgary, Alberta Canada T2L 2K7, 2008 edition, 2008.
- [5] M. S. Eldred, C. G. Webster, and P. G. Constantine. Evaluation of non-intrusive approaches for wiener-asky generalized polynomial chaos. *American Institute of Aeronautics and Astronautics*, 2008.
- [6] Jasmine Foo, Xiaoliang Wan, and George Em Karniadakis. The multi-element probabilistic collocation method (me-pcm): Error analysis and applications. *Journal of Computational Physics*, 2008.
- [7] Benjamin Ganis, Hector Klie, Mary F. Wheeler, Tim Wildey, Ivan Yotov, and Dongxiao Zhang. Stochastic collocation and mixed finite elements for flow in porous media. *Computer methods in applied mechanics and engineering*, 2008.
- [8] Thomas Gerstner and Michael Griebel. Numerical integration using sparse grids.
- [9] John Jakeman, Michael Eldred, and Dongbin Xiu. Numerical approach for quantification of epistemic uncertainty. *Journal of Computational Physics*, 2010.
- [10] G.J.A. Loeven, J.A.S. Witteveen, and H. Bijl. Probabilistic collocation: An efficient non-intrusive approach for arbitrarily distributed parametric uncertainties. *Aerospace Sciences Meeting*, 2007.
- [11] Sergey Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Doklady Akademii Nauk SSSR*, 1963.
- [12] Dongbin Xiu and George Em Karniadakis. Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos. *Computer methods in applied mechanics and engineering*, 2002.
- [13] Dongbin Xiu and George Em Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 2003.