(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 April 2010 (15.04.2010)

PCT

(10) International Publication Number
WO 2010/042746 A2

(51) International Patent Classification:
G06F 19/00 (2006.01)

(21) International Application Number:
PCT/US2009/060044

(22) International Filing Date:
8 October 2009 (08.10.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/104,154 9 October 2008 (09.10.2008) US

(71) Applicant *(for all designated States except US)*: **CHEVRON U.S.A. INC.** [US/US]; 6001 Bollinger Canyon Road, San Ramon, California 94583 (US).

(71) Applicant *(for CA only)*: **SCHLUMBERGER CANADA LIMITED** [CA/CA]; 525 - 3rd Avenue S.W., Calgary, Alberta T2P 0G4 (CA).

(71) Applicant *(for DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR only)*: **SERVICES PETROLIERS SCHLUMBERGER** [FR/FR]; 41, rue Saint Dominique, F-75007 Paris (FR).

(71) Applicant *(for all designated States except AL, AT, AU, BE, BG, BH, BR, CH, CL, CY, CZ, DE, DK, DZ, EG, ES, FR, GB, GR, IL, IS, KM, KP, LA, LC, LK, LR, LS, LY, MA, ME, MG, MN, MW, MZ, NA, NG, NI, PE, PG, PH, RS, SC, SG, SL, SM, ST, SV, SY, SZ, TZ, UG, US, VC, VN, ZM, ZW)*: **LOGINED B.V.** [NL/NL]; Gevers deynootweg 61, BJ Den Haag, NL-2586 Netherlands (NL).

(71) Applicant *(for all designated States except AE, AL, AM, AO, AT, AU, AZ, BE, BF, BG, BH, BJ, BR, BY, CA, CF, CG, CI, CL, CM, CZ, DE, DK, DO, DZ, EG, ES, FR, GA, GB, GN, GQ, GR, GW, HU, ID, IE, IL, IT, JP, KG, KZ, LV, MC, MD, ML, MR, MT, MX, MY, NE, NG, NL, PE, RU, SI, SN, SZ, TD, TG, TJ, TM, US, ZA)*: **PRAD RESEARCH AND DEVELOPMENT LIMITED** [GB/GB]; Craigmuir Chambers Road Town, Tortola (VG).

(71) Applicant *(for all designated States except US)*: **ETH ZURICH** [CH/CH]; Raemistrasse 101, ETH transfer, CH-8092 Zurich (CH).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **HAJIBEYGI, Hadi** [IR/CH]; Siewerdtstrasse 20, CH-8050 Zurich (CH). **BONFIGLI, Giuseppe** [IT/CH]; Faerberstrasse 6, CH-9000 St. Gallen (CH). **HESSE, Marc Andre** [DE/US]; 4510 W. Guadalupe, C303, Austin, Texas 78751 (US). **JENNY, Patrick** [CH/CH]; Fliweg 19, CH-8872 Weesen (CH).

(74) Agents: **NORTHCUTT, Christopher D.** et al.; Chevron Corporation - Law Department, 6001 Bollinger Canyon Road, San Ramon, California 94583 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM,

[Continued on next page]

(54) Title: ITERATIVE MULTI-SCALE METHOD FOR FLOW IN POROUS MEDIA

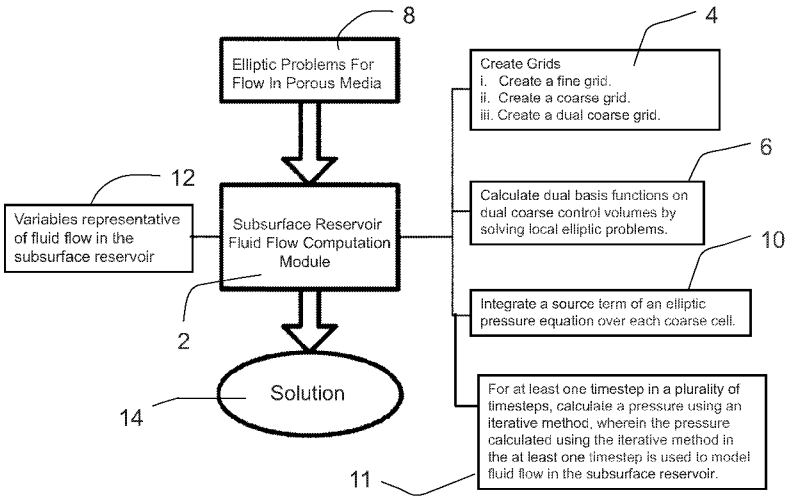


FIG. 1

(57) Abstract: Computer-implemented iterative multi-scale methods and systems are provided for handling simulation of complex, highly anisotropic, heterogeneous domains. A system and method can be configured to achieve simulation of structures where accurate localization assumptions do not exist. The iterative system and method smoothes the solution field by applying line relaxation in all spatial directions. The smoother is unconditionally stable and leads to sets of tri-diagonal linear systems that can be solved efficiently, such as by the Thomas algorithm. Furthermore, the iterative smoothing procedure, for the improvement of the localization assumptions, does not need to be applied in every time step of the computation.

WO 2010/042746 A2

AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

ITERATIVE MULTI-SCALE METHOD FOR FLOW IN POROUS MEDIA

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The present application for patent claims the benefit of provisional patent application United States Serial No. 61/104,154, filed October 9, 2008, which the entirety of the application is incorporated herein by reference.

TECHNICAL FIELD

[0002] The disclosure generally relates to computer-implemented simulators for characterizing fluid flow within subsurface formations, and more particularly, to computer-implemented simulators that use multi-scale methods to simulate fluid flow within subsurface formations.

BACKGROUND

[0003] Natural porous media, such as subterranean reservoirs containing hydrocarbons, are typically highly heterogeneous and complex geological formations. While recent advances, specifically in characterization and data integration, have provided for increasingly detailed reservoir models, classical simulation techniques tend to lack the capability to honor the fine-scale detail of these structures. Various multi-scale methods have been developed to deal with this resolution gap.

[0004] These multi-scale methods, which can be used for simulation of fluid flow in a subterranean reservoir, can be categorized into multi-scale finite-element (MSFE) methods, mixed multi-scale finite-element (MMSFE) methods, and multi-scale finite-volume (MSFV) methods. These methods aim to reduce complexity of the reservoir model by incorporating the fine-scale variation of coefficients into a coarse-scale operator. This is similar to

upscaling methods, which target coarse-scale descriptions based on effective, tensorial coefficients; however, multi-scale methods also allow for reconstruction of the fine-scale velocity field from a coarse-scale pressure solution. If a conservative fine-scale velocity field is obtained, which typically the MMSFE and MSFV methods can provide, the velocity field can then be used to solve the saturation transport equations on the fine grid. It will be appreciated by one skilled in that art, that for problems arising from flow and transport in porous media, a conservative velocity is desired for the transport calculations.

[0005] These multi-scale methods can be applied to compute approximate solutions at reduced computational cost. The multi-scale solutions can differ from the reference solutions that are computed with the same standard numerical scheme on the fine grid. While the permeability fields characterized by two separable scales typically converge with respect to coarse-grid refinement, these methods may not converge in the absence of scale separation due to error introduced by multi-scale localization assumptions. For instance, error introduced by a multi-scale method, with respect to the coarse cells, is typically prominent in the presence of large coherent structures with high permeability contrasts, such as nearly impermeable shale layers, where no general accurate localization assumption exists.

[0006] Multi-scale methods that are based on local numerical solutions of the fine-scale problem and thus honor the provided permeability field can be used to derive transmissibilities for the coarse problem. The quality of multi-scale results depends on the localization conditions employed to solve the local fine-scale problems. Previous methods have employed global information, such as an initial global fine-scale solution, to enhance the boundary conditions of the local problems. However, these methods may not provide value for fluid flow problems with high phase viscosity ratios, frequently changing boundary

conditions, or varying well rates. Other methods have iteratively improved the coarse-scale operator. For instance, the adaptive local-global (ALG) upscaling approach is based on global iterations to obtain a self-consistent coarse-grid description. Recently, ALG was also employed to improve the local boundary conditions in the multi-scale finite volume element method (ALG-MSFVE). While the ALG method has shown to be more accurate than local upscaling methods and leads to asymptotic solutions for a large number of iterations, the solutions typically can be different from standard fine-scale solutions and the error due to ALG can be problem dependent.

SUMMARY

[0007] Computer-implemented iterative multi-scale methods and systems are provided for simulation of anisotropic, heterogeneous domains. For example, a system and method can be configured to achieve simulation of structures where accurate localization assumptions cannot be made. The iterative method and system facilitates smoothing of the solution field by applying line relaxation in the spatial directions. The iterative smoothing procedure can be applied in fewer than all time steps of a computation. As an example, a system and method can include creating a fine grid, a coarse grid and a dual coarse grid, calculating dual basis functions on the dual coarse control volumes of the dual coarse grid by solving local elliptic problems, integrating a source term of an elliptic pressure equation over each coarse cell of the coarse grid, and for at least one timestep in a plurality of timesteps, calculating a pressure using an iterative method, where the pressure calculated using the iterative method in the at least one timestep can be used to model fluid flow in the subsurface reservoir.

[0008] As another example, a multi-scale computer-implemented method and system is provided for use in modeling fluid flow in a subsurface reservoir. The system and method

can include creating a fine grid defining a plurality of fine cells associated with a geological formation of the subsurface reservoir, a coarse grid defining a plurality of coarse cells having interfaces between the coarse cells, the coarse cells being aggregates of the fine cells, and a dual coarse grid defining a plurality of dual coarse control volumes, the dual coarse control volumes being aggregates of the fine cells and having boundaries bounding the dual coarse control volumes. In this example, the basis functions can be calculated on the dual coarse control volumes by solving local elliptic problems, and a source term of an elliptic pressure equation can be integrated over each coarse cell. The fine grid can be an unstructured grid.

[0009] For at least one timestep in a plurality of timesteps, the computation can include calculating a pressure using an iterative method. In an example, the iterative method can include, for each iteration: applying a smoothing scheme to a solution for the pressure over a fine grid from a previous iteration to provide a smoothed fine-grid pressure; calculating correction functions using the smoothed fine-grid pressure; applying a restriction operation that includes the correction functions to solve for the pressure over a coarse grid; and applying a prolongation operation to the pressure solved over the coarse grid to reconstruct an updated solution for the pressure over the fine-grid. The pressure calculated using the iterative method in the at least one timestep can be used to model fluid flow in a subsurface reservoir. In an example, the steps of the iterative method can be repeated until the solution for the pressure over the fine-grid converges.

[0010] In another example, a system and method can include, prior to calculating the pressure using the iterative method, a step of initializing the value for the pressure in the fine cells of a fine grid by setting that value equal to zero. The solution for the pressure over the fine grid in can be calculated using the calculated basis functions and the integrated source

term. In yet another example, a system and method can include re-computing the dual basis functions and the correction functions in a timestep and over the dual coarse control volumes where a change of the mobility coefficient of the local elliptic problems exceeds a predetermined threshold value.

[0011] The step of applying the smoothing scheme to the solution of the pressure over the fine grid can include applying a line-relaxation smoothing operation. Applying the line-relaxation smoothing operation can include: applying a linear operator that has a tri-diagonal structure to the solution of the pressure over the fine grid to provide a linear system of equations; and solving the linear system of equations using a Thomas algorithm.

[0012] A system and method can include outputting or displaying the pressure calculated using the iterative method in the at least one timestep.

[0013] As another example, a multi-scale computer-implemented method and system for use in modeling fluid flow in a subsurface reservoir can include computing a model using a finite volume method in a plurality of timesteps. The model can include one or more variables representative of fluid flow in the subsurface reservoir, where at least one of the one or more variables representative of fluid flow is responsive to calculated basis functions. The computing can include calculating basis functions on dual coarse control volumes of a dual coarse grid by solving local elliptic problems, integrating a source term of an elliptic pressure equation over each coarse cell of a coarse grid, and for at least one timestep of the plurality of timesteps, calculating a pressure using an iterative method. The results from the computed model, including the pressure calculated using the iterative method in the at least one timestep, can be used to model fluid flow in the subsurface reservoir.

[0014] The iterative method can include, for each iteration: applying a smoothing scheme to a solution for the pressure over a fine-grid from a previous iteration to provide a smoothed fine-grid pressure; calculating correction functions using the smoothed fine-grid pressure; solving for the pressure on a coarse grid using the correction functions; and reconstructing a solution for the pressure over the fine-grid using the result from solving for the pressure on the coarse grid. In an example, the smoothing scheme can include applying n_s smoothing steps, where n_s is a positive integer greater than 1. In another example, solving for the pressure on the coarse grid can include calculating the right-hand side of the linear system for the pressure over the coarse-grid using the correction functions, and solving for the pressure on the coarse grid using the calculated right-hand side of the linear system for the pressure over the coarse-grid. The steps of the iterative method can be repeated until the solution for the pressure over the fine-grid converges.

[0015] A system and method can include outputting or displaying the computed model including the pressure calculated using the iterative method in the at least one timestep.

[0016] In another example, a system and method can include re-computing the basis functions and the correction functions in the timestep and over the dual coarse control volumes where a change of the mobility coefficient of the local elliptic problems exceeds a predetermined threshold value. In another example, the system and method can include re-computing the correction functions in a timestep where the source term exceeds a predetermined limit.

[0017] A step of applying a smoothing scheme to the solution of the pressure over a fine grid can include applying a line-relaxation smoothing operation, including: applying a linear

operator that has a tri-diagonal structure to the solution of the pressure over the fine grid to provide a linear system of equations; and solving the linear system of equations using a Thomas algorithm.

[0018] A system for use in modeling fluid flow in a geological formation of a subsurface reservoir using a model is also provided. The system can including one or more data structures resident in a memory for storing data representing a fine grid, a coarse grid, a dual coarse grid, and dual basis functions calculated on the dual coarse control volumes by solving local elliptic problems; software instructions, for executing on one or more data processors, to compute the model using a finite volume method in at least two timesteps, and a visual display for displaying fluid flow in the geological formation of the subsurface reservoir using the computed model, including the pressure calculated using the iterative method in the at least two timestep. The model can include one or more variables representative of fluid flow in the subsurface reservoir, where at least one of the one or more variables representative of fluid flow is responsive to calculated basis functions. The computing can include calculating the basis functions on the dual coarse control volumes by solving local elliptic problems; integrating a source term of an elliptic pressure equation over each coarse cell; and for at least one timestep of the at least two timesteps, calculating a pressure using an iterative method.

[0019] The iterative method can include, for each iteration: applying a smoothing scheme to a solution for the pressure over the fine-grid from a previous iteration to provide a smoothed fine-grid pressure; calculating correction functions using the smoothed fine-grid pressure; solving for the pressure on the coarse grid using the correction functions; and reconstructing a solution for the pressure over the fine-grid using the result from solving for the pressure on the coarse grid.

[0020] As an illustration of an area of use for such techniques, the techniques can be used with a method for operating a subsurface reservoir to achieve improved production of a reservoir fluid (*e.g.*, oil) from a geological formation of the subsurface reservoir. For example, a system and method can include injecting a displacement fluid into a portion of the geological formation of the subsurface reservoir, and applying a reservoir fluid production process to the subsurface reservoir under at least one operational condition that is derived based on the results from executing the steps of any of the foregoing techniques. Nonlimiting examples of operational conditions are displacement fluid injection rate, reservoir fluid production rate, viscosity ratio of displacement fluid to reservoir fluid, location of injection of the displacement fluid, location of production of the reservoir fluid, displacement fluid saturations, reservoir fluid saturations, displacement fluid saturations at different pore volumes injected, and reservoir fluid saturations at different pore volumes injected.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] Figure 1 is a block diagram of an example computer structure for use in modeling fluid flow in a subsurface reservoir.

[0022] Figure 2 is a schematic view of a 2D grid domain showing an enlarged coarse cell.

[0023] Figure 3A is an illustration of a surface graph showing a 2D basis function.

[0024] Figure 3B is an illustration of a surface graph showing a 2D correction function.

[0025] Figure 4 is a flowchart illustrating steps used in a reservoir simulator employing an iterative multi-scale method.

[0026] Figure 5 is a flowchart illustrating multi-grid steps used in a reservoir simulator employing an iterative multi-scale method.

[0027] Figure 6A is a schematic view of a 2D grid showing a homogenous domain.

[0028] Figure 6B is a schematic view of a 2D heterogeneous mobility field.

[0029] Figure 7A is an illustration of numerical convergence histories in a homogeneous isotropic domain.

[0030] Figure 7B is an illustration of numerical convergence histories in a heterogeneous isotropic domain.

[0031] Figure 7C is an illustration of numerical convergence histories in a homogeneous anisotropic domain.

[0032] Figure 7D is an illustration of numerical convergence histories in a heterogeneous anisotropic domain.

[0033] Figure 8A is an illustration of convergence histories in a heterogeneous domain for $n_s = 10$ and various aspect ratios.

[0034] Figure 8B is an illustration of convergence rates in a heterogeneous domain for n_s as a function of aspect ratio.

[0035] Figure 9A is an illustration of convergence rates in a heterogeneous domain for various aspect ratios as a function of n_s.

[0036] Figure 9B is an illustration of effective convergence rates in a heterogeneous domain for various aspect ratios as a function of n_s.

[0037] Figure 10 is an illustration of convergence rates for various homogeneous –isotropic domain sizes.

[0038] Figure 11A is an illustration of convergence rates in a heterogeneous domain for a number of smoothing steps as a function of upscaling factors.

[0039] Figure 11B is an illustration of convergence rates in a heterogeneous domain for various aspect ratios as a function of upscaling factors.

[0040] Figures 12A – 12D are schematic views of a 2D domain showing a mobility field for various angles.

[0041] Figure 13 is a schematic view of a 2D domain showing wells having a source and sink of strength $q = \pm 1/(\Delta x \Delta y)$.

[0042] Figure 14A is an illustration of convergence rates in a domain for various aspect ratios and smoothing steps.

[0043] Figure 14B is an illustration of convergence rates in a domain for various upscaling factors and smoothing steps.

[0044] Figures 15A – 15D are illustrations of spectra in a homogeneous – isotropic domain for no-flow boundary conditions and various values of n_s.

[0045] Figure 16A is an illustration of eigenvectors with the largest eigenvalues of the spectra shown in Figure 15A.

[0046] Figure 16B is an illustration of corresponding residuum of the spectra shown in Figure 15A.

[0047] Figure 16C is an illustration of eigenvectors with the largest eigenvalues of the spectra shown in Figure 15C.

[0048] Figure 16D is an illustration of corresponding residuum of the spectra shown in Figure 15C.

[0049] Figures 17A and 17B are illustrations of spectra in a heterogeneous – isotropic domain for no-flow boundary conditions and various values of n_s.

[0050] Figure 18A is an illustration of corresponding residuum for eigenvectors with the ten largest eigenvalues of the spectra shown in Figure 16A.

[0051] Figure 18B is an illustration of corresponding residuum for eigenvectors with the ten largest eigenvalues of the spectra shown in Figure 16B.

[0052] Figure 19A is an illustration of the eigenvector with the largest eigenvalue of the spectra shown in Figure 17A.

[0053] Figure 19B is an illustration of the corresponding residuum of the spectra shown in Figure 17A.

[0054] Figures 20A and 20B are illustrations of permeability fields in a domain for the top and bottom layers of a 3D SPE10 test case.

[0055] Figures 21A and 21B are illustrations of convergence histories of the permeability fields shown in Figures 20A and 20B.

[0056] Figure 22 is an illustration of convergence histories of the permeability fields shown in Figure 20B.

[0057] Figure 23A is an illustration of a domain having two almost impermeable shale layers.

[0058] Figure 23B is an illustration of convergence histories of the permeability field of the domain shown in Figure 23A.

[0059] Figure 24A is an illustration of the fine-scale reference solution of a two-phase saturation map for the top layer of a 3D SPE10 test case.

[0060] Figure 24B is an illustration of the iterative multi-scale solution of a two-phase saturation map for the top layer of a 3D SPE10 test case.

[0061] Figure 24C is an illustration of the original multi-scale finite volume solution of a two-phase saturation map for the top layer of a 3D SPE10 test case.

[0062] Figure 25A is an illustration of the fine-scale reference solution of a two-phase saturation map for a domain having two almost impermeable shale layers as shown in Figure 23A.

[0063] Figure 25B is an illustration of the iterative multi-scale solution of a two-phase saturation map for a domain having two almost impermeable shale layers as shown in Figure 23A.

[0064] Figure 25C is an illustration of the original multi-scale finite volume solution of a two-phase saturation map for a domain having two almost impermeable shale layers as shown in Figure 23A.

[0065] Figure 26 illustrates an example computer system for use in implementing the methods.

DETAILED DESCRIPTION

[0066] Figure 1 depicts a block diagram of an example computer-implemented system for use in modeling fluid flow in a subsurface reservoir using a model. The system utilizes multi-scale physics to analyze fluid flow within the subsurface reservoir.

[0067] The system can include a computation module **2** for performing the computations discussed herein. The computation of the model can be performed at process **4** on a system of grids (*e.g.*, a fine grid, a coarse grid, and a dual coarse grid) as discussed in herein. Dual basis functions can be calculated at process **6** on the dual coarse control volumes of the dual coarse grid by solving local elliptic problems **8** for fluid flow in porous media. The model can include one or more variables **12** representative of fluid flow in the subsurface reservoir, wherein at least one of these variables is responsive to the calculated dual basis functions.

[0068] At process **10** in Figure 1, a source term of an elliptic pressure equation is integrated over each coarse cell, and at process **11**, for at least one timestep in a plurality of timesteps, a pressure is calculated using an iterative method, as discussed herein. The pressure calculated using the iterative method in the at least one timestep can be used to model fluid flow in the subsurface reservoir.

[0069] A multi-scale finite volume (MSFV) method is used for computing the model. Performance of the MSFV method can include calculating the dual basis functions on the dual control volumes of the dual coarse grid by solving elliptic problems (at process **6** of Figure 1).

[0070] A result of the computation can be a pressure that is used to model fluid flow in the subsurface reservoir, or a computed model comprising the pressure calculated using the iterative method in the at least one timestep that is used to model fluid flow in the subsurface reservoir.

[0071] The solution or result **14** of the computation can be displayed or output to various components, including but not limited to, a visual display, a user interface device, a computer readable storage medium, a monitor, a local computer, or a computer that is part of a network.

[0072] To explain an embodiment of a MSFV method, consider the elliptic problem

$$-\nabla \cdot (\boldsymbol{\lambda} \cdot \nabla p) = q \qquad \text{(Equation 1)}$$

on the domain Ω with the boundary conditions $\nabla p \cdot \boldsymbol{n} = f$ and $p(\boldsymbol{x}) = g$ at $\partial\Omega_1$ and $\partial\Omega_2$, respectively. Note that $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ is the whole boundary of the domain Ω and $\boldsymbol{n}$ is the outward unit normal vector. The mobility tensor $\boldsymbol{\lambda}$ is positive definite and the right-hand sides q, f and g are specified fields. The MSFV method in this embodiment is designed to efficiently compute approximate solutions of Equation (1) for highly heterogeneous coefficients $\boldsymbol{\lambda}$ and right-hand sides q, such as for mobility fields.

Such mobility fields depict a high variance, complex correlation structures and typically can be governed by a large range of length scales.

[0073] To illustrate further an MSFV technique, Figure 2 depicts a grid system which includes a fine-scale grid **100**, a conforming primal coarse-scale grid **200** shown in solid line, and a conforming dual coarse-scale grid **300** shown in dashed line. The fine-scale grid **100** is comprised of a plurality of fine cells **110**. The primal coarse-scale grid **200** has M primal coarse cells **210** and is constructed on the fine-scale grid **100** such that each primal coarse cell **210**, $\overline{\Omega}_k$ ($k \in [1,\ M]$), is comprised of multiple fine cells **110**. The dual coarse-scale grid **300**, which also conforms to the fine-scale grid **100**, can be constructed such that each dual coarse control volume or cell **310**, $\widetilde{\Omega}^h$ ($h \in [1,\ N]$), is comprised of multiple fine cells **110**. For example in Figure 2, both the primal coarse cells **210** and dual coarse cells **310** contain 11 × 11 fine cells. Each dual coarse cell **310** depicted in Figure 2 is defined by nodes **320**, $\boldsymbol{x}_k$, of the dual coarse-scale grid **300**. As illustrated in Figure 2, each primal coarse cell **210** ($\overline{\Omega}_k$) contains exactly one node **320**, $\boldsymbol{x}_k$, in its interior. Generally, each node **320** is centrally located in each primal coarse cell **210**. For example, the dual coarse-scale grid **300** can be constructed by connecting nodes **320** contained within adjacent primal coarse cells **210**. One skilled in the art will appreciate that the primal coarse-scale and dual coarse-scale grids can be much coarser than the underlying fine grid **300** on which the mobility field is represented. It is also emphasized that the multi-scale finite volume method is not limited to the grids shown in Figure 2, as very irregular grids or decompositions can be employed, as well as other sized grids such as the coarse cells containing 5 × 5 or 7 × 7 fine cells.

[0074] The reduction of degrees of freedom to describe the pressure field on the fine-scale grid **100** (fine pressure p_f) can be achieved through the approximation

$$p_f(\boldsymbol{x}) \approx p'(\boldsymbol{x}) = \sum_{h=1}^{N}\left[\sum_{k=1}^{M}\Phi_k^h(\boldsymbol{x})\overline{p}_k + \Phi^h(\boldsymbol{x})\right] \qquad \text{(Equation 2)}$$

where $\overline{p}_k$ are the pressure values at $\boldsymbol{x}_k$ nodes **320**. The basis functions can be represented as Φ_k^h and the correction function as Φ^h. Opposed to classic finite-element methods, the basis functions and correction functions are typically not analytical functions. For example, the basis functions and correction functions can be local numerical solutions of Equation (1) on $\widetilde{\Omega}^h$ without and with right-hand side, respectively. Localization can be achieved by employing reduced problem boundary conditions at $\partial\widetilde{\Omega}^h$, which is equivalent to

$$(\widetilde{\boldsymbol{n}}^h \cdot \nabla)\left((\boldsymbol{\lambda} \cdot \nabla\Phi_k^h) \cdot \widetilde{\boldsymbol{n}}^h\right) = 0 \qquad \text{(Equation 3)}$$

and

$$(\widetilde{\boldsymbol{n}}^h \cdot \nabla)\left((\boldsymbol{\lambda} \cdot \nabla\Phi^h) \cdot \widetilde{\boldsymbol{n}}^h\right) = r^h \qquad \text{(Equation 4)}$$

at $\partial\widetilde{\Omega}^h$ with $\widetilde{\boldsymbol{n}}^h$ being the unit normal vector pointing out of $\widetilde{\Omega}^h$. At the dual-grid nodes $\boldsymbol{x}_l$ which belong to $\widetilde{\Omega}^h$, $\Phi_k^h(\boldsymbol{x}_l) = \delta_{kl}$ and $\Phi^h(\boldsymbol{x}_l) = 0$. By construction, outside $\widetilde{\Omega}^h$, the Φ_k^h and Φ^h can be set to zero. An illustration of 2D basis and correction functions is shown in Figure 3A-B, where Figure 3A shows the 2D basis function and Figure 3B shows the 2D correction function.

[0075] To derive a linear system for the coarse pressure values $\overline{p}_k$, one can substitute p' of Equation (2) into Equation (1) and integrate over $\overline{\Omega}_l$, which leads to

$$-\int_{\overline{\Omega}_l} \nabla \cdot (\lambda \cdot \nabla p') d\Omega = -\int_{\overline{\Omega}_l} \nabla \cdot \left(\lambda \cdot \nabla \left(\sum_{h=1}^{N} \left(\sum_{k=1}^{M} \Phi_k^h \overline{p}_k + \Phi^h \right) \right) \right) d\Omega$$

$$= \int_{\overline{\Omega}_l} q d\Omega \qquad \text{(Equation 5)}$$

for all $l \in [1,\ M\ ]$. With the Gauss theorem one obtains

$$-\int_{\partial\overline{\Omega}_l} \left(\lambda \cdot \sum_{h=1}^{N} \left(\sum_{k=1}^{M} \overline{p}_k \nabla \Phi_k^h + \nabla \Phi^h \right) \right) \cdot \overline{\boldsymbol{n}}_l d\Gamma = \qquad \text{(Equation 6)}$$

$$\sum_{k=1}^{M} \overline{p}_k \sum_{h=1}^{N} \int_{\partial\overline{\Omega}_l} \left(-\lambda \cdot \nabla \Phi_k^h \right) \cdot \overline{\boldsymbol{n}}_l d\Gamma + \sum_{h=1}^{N} \int_{\partial\overline{\Omega}_l} \left(-\lambda \cdot \nabla \Phi^h \right) \cdot \overline{\boldsymbol{n}}_l d\Gamma = \int_{\overline{\Omega}_l} q d\Omega$$

which results in the linear system

$$A_{lk} \overline{p}_k = b_l \qquad \text{(Equation 7)}$$

for $\overline{p}_k$ with

$$A_{lk} = \sum_{h=1}^{N} \int_{\partial\overline{\Omega}_l} \left(-\lambda \cdot \nabla \Phi_k^h \right) \cdot \overline{\boldsymbol{n}}_l d\Gamma \qquad \text{(Equation 8)}$$

and

$$b_l = \int_{\overline{\Omega}_l} q d\Omega - \sum_{h=1}^{N} \int_{\partial\overline{\Omega}_l} \left(-\lambda \cdot \nabla \Phi^h \right) \cdot \overline{\boldsymbol{n}}_l d\Gamma \qquad \text{(Equation 9)}$$

The unit normal vector $\overline{\boldsymbol{n}}_l$ points out of $\overline{\Omega}_l$. Note that the right-hand side b_l also contains the effects of the fine-scale fluxes across $\partial\overline{\Omega}_l$ induced by the correction functions Φ^h.

[0076] With $\overline{p}_k$ and the superposition of Equation (2) one obtains the fine-scale pressure p', which is an approximation of the fine-scale reference solution p_f. In the MSFV method the difference between p' and p_f can be solely due to the localization assumption of Equation (4), *i.e.*, with

$$r^h = \left(\widetilde{\boldsymbol{n}}^h \cdot \nabla \right) \left(\left(\boldsymbol{\lambda} \cdot \nabla p_f \right) \cdot \widetilde{\boldsymbol{n}}^h \right) \text{ at } \partial\widetilde{\Omega}^h \quad \forall h \in [1, N] \qquad \text{(Equation 10)}$$

the two fine-scale pressure fields become identical.

[0077] For a wide range of examples, the MSFV method with $r^h = 0$ can lead to accurate results. In other words, the reduced problem boundary conditions typically provide a good localization assumption.

[0078] For multiphase problems, a conservative fine-scale velocity field can maintain mass-balance of the transported phase saturations. The velocity

$$\boldsymbol{u}' = -\boldsymbol{\lambda} \cdot \nabla p' \qquad \text{(Equation 11)}$$

fulfills this requirement in a weak sense, that is, while it is conservative for each coarse volume $\overline{\Omega}_k$, it is not conservative at the fine-scale. Therefore, a further step can be applied to solve saturation transport on the fine grid. To reconstruct a conservative fine-scale velocity field $\boldsymbol{u}''$, which is consistent with $\boldsymbol{u}'$, the additional local problems

$$-\nabla\cdot\left(\boldsymbol{\lambda}\cdot\nabla p_k''\right)=q \quad \text{on} \quad \overline{\Omega}_k \qquad \text{(Equation 12)}$$

with

$$\left(\boldsymbol{\lambda}\cdot\nabla p_k''\right)\cdot\overline{\boldsymbol{n}}_k=\left(\boldsymbol{\lambda}\cdot\nabla p'\right)\cdot\overline{\boldsymbol{n}}_k \quad \text{at} \quad \partial\overline{\Omega}_k \qquad \text{(Equation 13)}$$

are solved. The velocity field

$$\boldsymbol{u}''=\begin{cases}\boldsymbol{\lambda}\cdot\nabla p_k'' & \text{on} \quad \overline{\Omega}_k\\ \boldsymbol{\lambda}\cdot\nabla p' & \text{at} \quad \partial\overline{\Omega}_k\end{cases} \qquad \text{(Equation 14)}$$

for all $k\in[1,\ M\ ]$ is conservative (provided p'' is obtained with a conservative scheme) and can be employed to solve transport equations on the fine grid. For multiphase subsurface flow problems, for example, saturation transport may be calculated explicitly or implicitly. Since the mobility $\boldsymbol{\lambda}$ can depend on the saturations in the implicit version, iterations can be performed between the pressure Equation (1), which is solved with the MSFV method, and the transport equations. Good efficiency can be achieved when the transport equations are solved implicitly on the individual domains $\overline{\Omega}_k$. A Schwarz overlap scheme can then be employed to couple the local solutions. With this technique, which can be very efficient for

hyperbolic problems, the low computational complexity of the overall MSFV method can be maintained for multiphase flow.

[0079] The MSFV method can be adaptive, which provides several benefits to the user. For example, the conservative velocity reconstruction described above can be required in coarse cells $\overline{\Omega}_k$ where fine-scale transport is of interest. Moreover, the basis and correction functions can advantageously be stored and reused for subsequent time steps. The basis and correction functions can be recomputed in dual cells $\widetilde{\Omega}^h$ where changes of the coefficient λ exceeds a specified limit. Additionally, the correction functions can be recomputed when the right-hand side q exceeds a specified limit. In order to make the MSFV method more applicable for simulation of fluid flow within a subterranean reservoir, the MSFV method can be extended to cover factors such as compressibility, gravity, and complex well schemes. These extended versions of the MSFV method can prove to be effective for computation of a wide range of examples for which the multi-scale and fine-scale solutions are in agreement.

[0080] An iterative MSFV (i-MSFV) method is discussed. As already pointed out, the difference between the MSFV solution p' and the fine scale reference pressure p_f can be due to the localization assumptions, such that p' and p_f become identical if the boundary conditions obtained through Equation (4) are employed in fulfillment of r^h which is given in Equation (10). However, Equation (10) can require *a priori* knowledge of p_f.

[0081] A convergent iterative procedure to improve the localization boundary conditions, which does not depend on p_f, can be used. Therefore, the iterative improvement of r^h can be written as

$$r^{h^{(t)}} = \left(\tilde{\boldsymbol{n}}^h \cdot \nabla\right)\left(\left(\boldsymbol{\lambda} \cdot \nabla p_s'^{(t)}\right) \cdot \tilde{\boldsymbol{n}}^h\right) \text{ at } \partial\tilde{\Omega}^h \quad \forall h \in [1, N] \qquad \text{(Equation 15)}$$

The superscript (t) denotes the iteration level and

$$p_s'^{(t)} = S^{n_s} \cdot p'^{(t)} = S^{n_s} \cdot \sum_{h=1}^{N}\left[\sum_{k=1}^{M} \Phi_k^h \bar{p}_k^{(t)} + \Phi^{h^{(t-1)}}\right] \qquad \text{(Equation 16)}$$

is the smoothed MSFV fine-scale pressure approximation, where S is a linear smoothing operator and n_s represents the number of smoothing steps. One skilled in the art will recognize that the correction functions $\Phi^{h^{(t-1)}}$ can be based on the local boundary conditions of Equation (4) with $r^h = r^{h^{(t-1)}}$.

[0082] For a more compact presentation of the iterative MSFV (i-MSFV) method, the fine-grid values of p_s', p', Φ_k^h, and Φ^h can be ordered in vectors $\boldsymbol{p}_s'$, $\boldsymbol{p}'$, $\boldsymbol{\Phi}_k^h$, and $\boldsymbol{\Phi}^h$ with entries $\left[p_s'\right]_i$, $\left[p'\right]_i$, $\left[\Phi_k^h\right]_i$, and $\left[\Phi^h\right]_i$, respectively. The linear equations involved in this iterative procedure can be expressed in matrix form by writing

$$\left[\Phi^{h^{(t-1)}}\right]_i = C_{ij}^h \left[p_s'^{(t-1)}\right]_j + E_i^h \qquad \text{(Equation 17)}$$

$$A_{lk}\,\overline{p}_k^{(t)} = \underbrace{Q_l}_{\int_{\overline{\Omega}_l} q\,d\Omega} + \underbrace{D_{li}\left[\Phi^{h\,(t-1)}\right]_i}_{-\sum_{h=1}^{N}\int_{\partial\overline{\Omega}_l}\left(-\lambda\cdot\nabla\Phi^{h(t-1)}\right)\cdot\overline{\boldsymbol{n}}_{\boldsymbol{l}}\,d\Gamma} \qquad \text{(Equation 18)}$$

$$\left[p_s'^{(t)}\right]_i = \left[S^{n_s}\right]_{ij}\sum_{h=1}^{N}\left(\left[\Phi_k^h\right]_j\,\overline{p}_k^{(t)} + \left[\Phi^{h(t-1)}\right]_j\right) \qquad \text{(Equation 19)}$$

Equation (17) corresponds to the localized problems for the correction functions $\Phi^{h^{(t-1)}}$, as following from the original elliptic equation of Equation (1) with the boundary condition of Equation (4) defined according to Equation (15). The terms on the right-hand side express the linear dependence of $\Phi^{h^{(t-1)}}$ on the smoothed pressure field $p_s'^{(t-1)}$ at the previous iterative step (due to the iterative boundary condition of Equation (15)) and on the source term q of the elliptic problem, respectively. Equation (18) corresponds to the coarse-scale problem of Equation (5), and is equivalent to Equations (7) - (9). Finally, Equation (19) expresses the iterative reconstruction formula of Equation (16). Combining Equations (17), (18), and (19) and introducing the identity matrix $\boldsymbol{I}$, the following linear relation can be obtained:

$$\left[p_s'^{(t)}\right]_i = \underbrace{\left[S^{n_s}\right]_{ij}\sum_{h=1}^{N}\left(\left[\Phi_k^h\right]_j A_{kl}^{-1}\left(Q_l + D_{lq}E_q^h\right) + E_j^h\right)}_{b_i^*}$$

$$+ \underbrace{\left[S^{n_s}\right]_{ij} \sum_{h=1}^{N} \left[\left(\left[\Phi_k^h\right]_j A_{kl}^{-1} D_{lq} + I_{jq}\right) C_{qr}^h\right]}_{A_{iq}^{*(n_s)}} \left[p_s'^{(t-1)}\right]_r} \quad \text{(Equation 20)}$$

between the smoothed fine-scale pressure fields $p_s'^{(t-1)}$ and $p_s'^{(t)}$, at two consecutive iteration steps.

[0083] A process flowchart of the i-MSFV method **400** is shown in Figure 4. The fine-scale pressure is initialized in step **410**. For example, in step **410** the fine-scale pressure can be set to zero. Basis functions are computed in step **420** and the right-hand side of the elliptic pressure equation is integrated over each coarse volume in step **430**. One skilled in the art will appreciate that these steps can be performed once and can then be followed by the main iteration loop. At the beginning of each iteration of step **440**, n_s smoothing steps **441** can be applied, and the smoothed fine-scale pressure is employed to compute the correction functions in step **443**. The correction functions are used to obtain the right-hand side of the linear system for the coarse pressure, which is shown in step **445**. At the end of each iteration, the coarse system is solved in step **447** and the new fine-scale pressure approximation is reconstructed in step **449**. The components of the vector $\bar{\boldsymbol{p}}$ can be the actual pressure values at the dual coarse-grid nodes. This process can be expressed in algorithmic form such that it can be employed for use in a reservoir simulator to simulate fluid flow within a subterranean reservoir:

initialize $p'^{(t=0)}$

$\forall$ h : $\forall$ k : compute basis functions $\boldsymbol{\Phi}_k^h$

calculate $\boldsymbol{Q}$; Equation (18)

for $t = 1$ to number of i-MSFV iterations {

$\boldsymbol{p}_s'^{(t-1)} = \boldsymbol{p}'^{(t-1)}$

for $i = 1$ to n_s {

$\boldsymbol{p}_s'^{(t-1)} = \boldsymbol{S} \cdot \boldsymbol{p}'^{(t-1)}$; smoothing step

}

$\forall$ h : compute correction function $\Phi^{h^{(t-1)}}$; based on $\boldsymbol{p}_s'^{(t-1)}$

calculate $\boldsymbol{b}^{(t-1)} = \boldsymbol{Q} + \boldsymbol{DC} \cdot \boldsymbol{p}_s'^{(t-1)} + \boldsymbol{DE}$; Equation (18)

solve coarse system $\boldsymbol{A} \cdot \overline{\boldsymbol{p}}^{(t)} = \boldsymbol{b}^{(t-1)}$; Equation (18)

reconstruct $\boldsymbol{p}'^{(t)}$; Equation (2)

}

[0084] The i-MSFV method can be interpreted as a multi-grid method, where differential equations are solved using a hierarchy of discretizations. The i-MSFV method can include smoothing, restriction, and prolongation steps which are typical operations employed in multi-grid methods. For example, the i-MSFV method can include a smoothing step to improve the approximate fine-grid solution $\boldsymbol{p}'^{(t-1)}$, a restriction step to obtain the right-hand side $\boldsymbol{b}^{(t-1)}$ of the coarse grid system that can be used for solving the pressure values $\overline{\boldsymbol{p}}^{(t)}$, and a prolongation step to acquire the updated fine-grid solution.

[0085] In Figure 5, the i-MSFV method **500**, schematically depicted as a multi-grid method, includes applying n_s smoothing steps **510** to improve the approximate fine-grid pressure solution $\boldsymbol{p}'^{(t-1)}$. For example, smoothing steps **510** can be performed by an iterative linear solver. The smoothed fine-grid pressure can be employed to update the correction functions $\Phi^{h^{(t-1)}}$. One skilled in the art will appreciate that the coarse-grid operator $\boldsymbol{A}$, which is based on the basis functions $\boldsymbol{\Phi}_k^h$, can be constructed a single time; whereas, the correction functions $\Phi^{h^{(t-1)}}$ can undergo iterative refinement. The updated correction functions $\Phi^{h^{(t-1)}}$ can be used in a subsequent restriction step **520** to obtain the right-hand side of the linear system for the coarse pressure. In particular, restriction step **520** leads to the right-hand side $\boldsymbol{b}^{(t-1)}$ of the coarse grid system, which can be used for solving the pressure values $\overline{\boldsymbol{p}}^{(t)}$ at the dual coarse-grid nodes in step **530**. For example, the coarse system can be solved with any suitable solver by solving $\overline{\boldsymbol{p}}^{(t)} = \boldsymbol{A}^{-1}\boldsymbol{b}^{(t-1)}$. Due to the typically extreme coarsening factors, the coarse problem can be small enough to be solved directly. Once the coarse system is solved, the updated fine-grid pressure solution $\boldsymbol{p}'^{(t)}$ can be obtained through prolongation shown in step **540**. Prolongation can be achieved simply by superimposing the correction functions plus the basis functions weighted with the new coarse pressure values. The reconstructed approximation of the fine-grid pressure **550**, $\boldsymbol{p}'^{(t)}$, can then be used for the next iteration $t = t + 1$ shown in step **560**.

[0086] Although shown here for two grid levels, the i-MSFV method can be extended for more complex cycles. Moreover, it can be seen from the operations in Figures 4 and 5 that

no assumptions regarding the topologies of the fine-scale and coarse-scale grids are made. For example, the same methodology can be applied for unstructured fine grids, and instead of coarse grids one can employ appropriate domain decompositions. The smoothing scheme can be utilized to achieve robustness and good convergence. For example, consistent line-relaxation works very well as a smoothing scheme due to the effectiveness of line-relaxation to distribute the residuum from the coarse-scale dual cell boundaries across the domain. Moreover, line-relaxation can depend weakly on the grid aspect ratio and on the level of anisotropy. One skilled in the art will appreciate that a different smoothing scheme can be used for unstructured fine grids.

[0087] Regarding fine-scale smoothing, the following is provided. As already mentioned herein, line-relaxation (LR) is one possibility to smooth the approximate fine-grid solution $\boldsymbol{p}'^{(t-1)}$. However, other smoothing schemes can be used. To illustrate a smoothing scheme, line-relaxation can be employed in an example i-MSFV implementation. Therefore, consider the fine-scale system

$$\boldsymbol{M} \cdot \boldsymbol{p}_f = \boldsymbol{R} \qquad \text{(Equation 21)}$$

which results form a conservative finite-volume discretization of Equation (1) on the fine grid. For simplicity, assume that the grid lines are parallel to the *x*-, *y*- and *z*-directions of a Cartesian coordinate system. The linear operator can be split as $\boldsymbol{M} = \boldsymbol{M}_x + \boldsymbol{M}_y + \boldsymbol{M}_z$, where $\boldsymbol{M}_{x,y,z}$ represent the discretizations of the elliptic operator in the corresponding coordinate directions. If one operator plus the diagonal components of the other ones are treated implicitly, the following iterative scheme is obtained:

$$\left(\boldsymbol{M}_x + diag\left(\boldsymbol{M}_y + \boldsymbol{M}_z\right)\right)\cdot \boldsymbol{p}^{\upsilon+1/3} = \boldsymbol{R} - \left(\boldsymbol{M}_y + \boldsymbol{M}_z - diag\left(\boldsymbol{M}_y + \boldsymbol{M}_z\right)\right)\cdot \boldsymbol{p}^{\upsilon}$$

(Equation 22)

$$\left(\boldsymbol{M}_y + diag\left(\boldsymbol{M}_x + \boldsymbol{M}_z\right)\right)\cdot \boldsymbol{p}^{\upsilon+2/3} = \boldsymbol{R} - \left(\boldsymbol{M}_x + \boldsymbol{M}_z - diag\left(\boldsymbol{M}_x + \boldsymbol{M}_z\right)\right)\cdot \boldsymbol{p}^{\upsilon+1/3}$$

(Equation 23)

$$\left(\boldsymbol{M}_z + diag\left(\boldsymbol{M}_x + \boldsymbol{M}_y\right)\right)\cdot \boldsymbol{p}^{\upsilon+1} = \boldsymbol{R} - \left(\boldsymbol{M}_x + \boldsymbol{M}_y - diag\left(\boldsymbol{M}_x + \boldsymbol{M}_y\right)\right)\cdot \boldsymbol{p}^{\upsilon+2/3}$$

(Equation 24)

where $\boldsymbol{p}^{\upsilon}$ is the approximate solution after the $\upsilon - \text{th}$ line-relaxation step and $diag\left(\boldsymbol{M}_x\right)$ represents the matrix with the diagonal of $\boldsymbol{M}_x$. In this scheme, the three linear systems (22) - (24) are solved sequentially at each iteration. For a two-point flux approximation, the linear operators $\boldsymbol{M}_{x,y,z}$ have a tri-diagonal structure and the systems (22) - (24) can be solved, for example with the Thomas algorithm, which has a linear complexity. Moreover, the three linear operators can further be split into independent linear systems for each grid line. This property can be useful for massive parallel computing, which can be advantageously used in the field of reservoir simulation. This iterative line-relaxation solver can be convergent, but for big problems the rate can be slow. In this multi-scale framework, however, few line-relaxation steps can be applied to smooth $\boldsymbol{p}'^{(t-1)}$ sufficiently for an effective improvement of the local boundary conditions. The optimum number of smoothing steps per i-MSFV iteration can be case dependent.

EXAMPLE NUMERICAL RESULTS

[0088] The convergence rate of the i-MSFV method can be assessed. The first set of examples discussed below are based on a test case consisting of a rectangular 2D domain

with constant pressure and no-flow conditions at the vertical and horizontal boundaries, respectively. For the discretization, an equidistant Cartesian fine grid with 44 × 44 cells was used and in addition, for the i-MSFV method, a 4 × 4 coarse grid was employed (Figure 6A). Since each coarse cell is composed of 11×11 fine cells, the upscaling factor is 11 in each coordinate direction. For the following examples, homogeneous and heterogeneous mobility fields and domains with various aspect ratios α (horizontal to vertical dimension) are considered. The size of each fine cell is $\Delta x \times \Delta y$ with $\Delta x = 1 = \alpha \Delta y$. One skilled in the art will appreciate that a case with isotropic mobility and $\Delta x = \alpha \Delta y$ is numerically identical to a case with $\Delta x = \Delta y$ and a mobility which is larger by a factor of α^2 in the y-direction.

[0089] The homogeneous examples with $\lambda_{ij} = \delta_{ij}$ also include a source with $q = 1/(\Delta x \Delta y)$ and a sink with $q = -1/(\Delta x \Delta y)$ distributed over the fine cells; the fine cells are depicted in black in Figure 6A. For the heterogeneous cases, the mobility field that is depicted in Figure 6B, which has a natural logarithm (*ln*) variance of 6.66 and mean of −0.29, was used. These cases are a part of the top layer of a three-dimensional SPE10 test case [M.A. Christie and M.J. Blunt. Tenth SPE comparative solution project: A comparison of upscaling techniques. SPE 66599, presented at *The SPE Symposium on Reservoir Simulation*, Houston, TX, February, 2001.].

[0090] Figure 7 shows the base-10 logarithm (*log*) of the maximum error in the domain, i.e. *log*(ε) with $\varepsilon = \left\| p' - p_f \right\|_\infty$, as a function of i-MSFV iterations and smoothing steps (per iteration), n_s, for the homogeneous (Figures 7A and 7C) and heterogeneous (Figures 7B and

7D) cases with α = 1 (Figures 7A and 7B) and α = 10 (Figures 7C and 7D). For the cases there exists a minimum n_s, for which the i-MSFV method converges. The best convergence can be observed for the homogeneous isotropic (α = 1) case and the worst convergence for the heterogeneous case with α = 10.

[0091] Figure 8A shows the convergence histories for the heterogeneous test case as a function of α with n_s = 10. The slope decreases as α increases, but eventually it approaches an asymptotic value. This observation is confirmed by the plot in Figure 8B, which shows the convergence rate (average slope between $log(\varepsilon) = -2$ and $log(\varepsilon) = -8$) as a function of α and n_s for the heterogeneous case. For $\alpha \geq 20$, the convergence dependence on the aspect ratio becomes negligible. This demonstrates that the i-MSFV method can be applied for cases with very large aspect ratios and/or extreme anisotropies. For comparison, the convergence rates (multiplied with 100) are also shown for the line relaxation method, which can be employed as a smoother within the i-MSFV algorithm, shown in Figure 8B.

[0092] Figure 9A illustrates how the convergence rate increases with n_s. To estimate the optimal number of smoothing steps per iteration, the assumption is made that the amount of computational work to calculate the correction functions, to solve the coarse problem, and to reconstruct p' corresponds to β times the computational work required for one smoothing step. This leads to the relation:

$$\text{Effective Convergence Rate} = \frac{\text{Convergence Rate}}{1 + n_s/\beta} \qquad \text{(Equation 25)}$$

This can be a measure for the error reduction, if the computational work equivalent to one MSFV iteration (without smoothing nor reconstruction of a conservative velocity field) is invested. Figure 9B shows the effective convergence rates for various aspect ratios α as functions of n_s, where β is assumed to be one.

[0093] To analyze the computational cost associated with the i-MSFV method as a function of the problem size, the number of fine cells in the homogeneous isotropic test case was increased successively by adding coarse-grid cells each comprised of 11×11 fine cells. Figure 10 depicts the convergence rates for 2×2, 3×3, 4×4, 5×5, 6×6, 7×7, 8×8, 9×9, and 10×10 coarse grids. The *log−log* plot shows that the convergence rate (for constant n_s = 10) for the i-MSFV method, which is indicated by a dashed line, can be insensitive to the fine-grid size in comparison to the convergence rate of the line relaxation solver, which is indicated by a solid line. Moreover, calculation steps in the i-MSFV algorithm, other than solving the global coarse-scale problem, can be performed locally and independently. Therefore, since up to very large cases the cost for solving the coarse system can be virtually negligible, the i-MSFV method can be a very efficient linear solver for large, stiff problems, and it can be used for massive parallel computing which can be employed in the field of reservoir simulation.

[0094] Another parameter of interest is the upscaling factor. Figures 11A-B show the convergence rate for the heterogeneous case with upscaling factors Γ of 11 × 11, 7 × 7, and 5 × 5. In Figure 11A, the convergence rates are shown as functions of α with constant n_s = 10 and in Figure 11B, they are depicted as functions of n_s with constant α = 5. One skilled in the art will appreciate that the optimal choice of Γ depends on the size of the fine grid and

on the computational cost of the individual algorithmic components. Therefore, the optimal choice of Γ depends highly upon the coarse-scale solver .

[0095] Four sets of 20 realizations of log-normally distributed mobility fields with spherical variogram and dimensionless correlation lengths ψ_1 = 0.5 and ψ_2 = 0.02 are generated using sequential Gaussian simulations. For each set, variance and mean of $ln(\lambda)$ are 2.0 and 3.0, respectively. As depicted in Figure 12, the angles θ between the long correlation length and vertical domain boundaries (or vertical grid lines orientation) are 0° (Figure 12A), 15° (Figure 12B), 30° (Figure 12C), and 45° (Figure 12D). For each case, a 100 × 100 fine grid and a 20 × 20 coarse grid was employed. At the boundaries of the quadratic domain, no-flow conditions were applied and at the lower left and upper right corners (cells (3,3) and (97,97)), a source and a sink of equal strength ($q = \pm 1/(\Delta x \Delta y)$) were imposed; these cells are indicated by a black dot in Figure 13. Figures 14A and 14B show the mean convergence rates as functions of θ for various n_s, α, and Γ. For example, Figure 14A depicts α = 1 and α = 5 and Figure 14B depicts a 5 × 5 and 7 × 7 upscaling factor. As shown in Figures 14A and 14B, there can be a significant difference in the convergence rates. However, in general, the convergence rate can decrease with increasing layering orientation angle θ.

[0096] Regarding spectral analysis of the i-MSFV method, the following are discussed. The convergence assessment of the i-MSFV method may also be observed by analyzing the spectrum of the associated iteration matrix, *i.e.*, according to Equation (20), of $A^{*(n_s)}$ in

$$p_s'^{(t)} = A^{*(n_s)} \cdot p_s'^{(t+1)} + b^* \qquad \text{(Equation 26)}$$

The iteration procedure converges if eigenvalues of $A^{*(n_s)}$ lay within the unit-disc of the complex plane.

[0097] Various spectra of $A^{*(n_s)}$ for the homogeneous anisotropic test case with no flow conditions at all boundaries are depicted in Figures 15A – 15D. Fine and coarse grids consist of 44 × 44 and 4 × 4 cells, respectively, and Figures 15A – 15D refer to iteration matrices based on various n_s. In particular, Figure 15A is for n_s = 0, Figure 15B is for n_s = 1, Figure 15C is for n_s = 2, and Figure 15D is for n_s = 5. These results confirm those presented in Figure 7A, such that at least two smoothing steps may be required for convergence of the homogeneous-isotropic case. It is noted that, unlike the matrix M of the fine-scale problem of Equation (21), $A^{*(n_s)}$ is not symmetric and possesses non-real eigenvalues. Eigenvalues of $A^{*(n_s)}$ are clustered around the negative real axis, which implies that the approximate solution at successive iteration steps oscillates around the exact one.

[0098] The eigenfunctions $\tilde{p}$ associated with the largest eigenvalues are plotted in Figures 16A-D together with the corresponding residuum $\rho = \nabla \cdot \lambda \cdot \nabla \tilde{p}$ in the discrete fulfillment of Equation (1) without right-hand side. Figure 16A is an illustration of eigenvectors with the largest eigenvalues of the spectra shown in Figure 15A. Figure 16B is an illustration of corresponding residuum in the fulfillment of Equation 1 without the right hand side of the spectra shown in Figure 15A. Figure 16C is an illustration of eigenvectors

with the largest eigenvalues of the spectra shown in Figure 15C. Figure 16D is an illustration of corresponding residuum in the fulfillment of Equation 1 without the right hand side of the spectra shown in Figure 15C. Only the results for $n_s = 0$ (unstable) and $n_s = 2$ (stable) are shown. In both cases, the residuum is largest at the dual-cell boundaries and without smoothing it is zero everywhere else. This is in agreement with the fact that any non-smoothed solution p' fulfills Equation (1) exactly inside the coarse dual cells. The smoothing steps efficiently redistribute the residuum and reduce its maximum amplitude. Consequently, the eigenvectors of $A^{*(n_s)}$ become amplified for $n_s = 0$ and are damped for $n_s = 2$.

[0099] In Figures 17A and 17B, spectra of the iteration matrix $A^{*(n_s)}$ can be observed for cases with heterogeneous isotropic mobility fields with no flow boundary conditions. Again, fine and coarse grids consist of 44 × 44 and 4 × 4 cells, respectively. Figure 17A is for $n_s = 0$ and Figure 17B is for $n_s = 5$. In Figures 18A and 18B, the largest values of the residua associated with the ten least stable eigenvectors for $n_s = 0$ and $n_s = 5$, respectively, are presented. Notice the discontinuous distribution in Figure 18A for the case with $n_s = 0$. The residuum gets distributed by the $n_s = 5$ smoothing steps (Figure 18B). Finally, Figures 19A and 19B depict the least stable eigenvector for $n_s = 5$ and its residuum.

[00100] Following is a discussion of application to subsurface flow. In typical incompressible subsurface reservoir flow simulations, the pressure in the porous media is governed by Equation (1). As in the examples herein, the mobility λ typically has a

complex distribution with high variance and sharp contrasts. It can be a function of the rock permeability $\boldsymbol{k}$, the fluid phase saturations and the fluid viscosities. For single-phase flow of a fluid with viscosity μ one can write $\boldsymbol{\lambda} = \boldsymbol{k} / \mu$. The expression for multiphase flow can be based on the relative permeability concept and can be expressed as $\boldsymbol{\lambda} = \boldsymbol{k} \sum_{j=1}^{n_p} k_{r_j} / \mu_j$ (where n_p is the number of fluid phases). The relative permeabilities k_{r_j} can be specified for each fluid phase j as functions of the saturations. While $\boldsymbol{\lambda}$ does not change with time in single-phase flow simulations, it evolves if multiple fluid phases are transported through the reservoir. For the following examples, the right-hand side of Equation (1) is non-zero only at the well. Therefore, no capillary pressure difference between the fluid phases and no gravity are considered.

[00101] Regarding single-phase flow, the following are discussed. The convergence behavior of the i-MSFV method for single-phase flow in particularly challenging reservoirs can be investigated in the following examples. The rectangular 2D domain can be discretized by a Cartesian, equidistant 220 × 55 fine-scale grid. No-flow conditions are applied at the bottom and top walls; at the left and right boundaries constant dimensionless pressure values of 1 and 0 are applied, respectively. Permeability fields from the top and bottom layers of the 3D SPE 10 test case are shown in Figures 20A and 20B and the corresponding convergence histories for the permeability fields are shown in Figures 21A and 21B, respectively, where a 20×5 coarse grid was employed. As with previous examples, the error can be defined as the logarithm of the maximum absolute difference between the approximate i-MSFV and the reference fine-scale pressure values. While for the top layer a good convergence rate can be

achieved with n_s = 10 (Figure 21A), approximately 250 smoothing steps can be applied for optimal convergence with the bottom layer permeability field (Figure 21B). However, in this example, many more smoothing steps can be applied if line relaxation is employed as an iterative linear solver (~10^5 iterations can be performed to reduce the error by five orders of magnitude). Moreover, Figure 22 illustrates that the number of smoothing steps can be reduced dramatically, if a coarsening factor of Γ = 5 × 5 (and fine grid of 220 × 60) instead of Γ = 11 × 11 (and fine grid of 220 × 55) is employed.

[00102] As a further example, a rectangular domain with two almost impermeable shale layers is considered (Figure 23A); the mobility in the shale layers is 10^{10} times smaller than in the rest of the domain. The equidistant Cartesian fine grid consists of 55 × 55 cells and the coarse grid for the i-MSFV method contains 5 × 5 volumes. Again, no-flow conditions are applied at the bottom and top boundaries and at the left and right sides the dimensionless pressure values are set equal to 1 and 0, respectively. Figure 23B shows the convergence histories with n_s = 10 for various aspect ratios.

[00103] Regarding multi-phase flow, the following are discussed. As already pointed out, in multiphase flow simulations the mobility λ and therefore the pressure field evolve with time as the phase saturations are transported through the domain. One skilled in the art will recognize that this also affects the localization boundary conditions, which continuously experience changes in the whole domain even where the mobility remains constant. Consequently, in a straight forward application of the i-MSFV method for multiphase flow, correction functions can be re-computed multiple times for each time-step. Although the number of i-MSFV iterations can be reduced by the good initial condition obtained from the

previous time step, such an approach can be more expensive than the original MSFV method. However, infrequently updating the localization boundary conditions for the re-computation of the correction functions can be sufficient to obtain highly accurate solutions. While in this example a converged solution is computed at the beginning of a simulation, the same localization boundary conditions can be used for a number of subsequent time steps and can be updated infrequently, such as each tenth time step by applying a single iteration. Therefore, for the major part of the simulation, the original MSFV method with slightly modified correction functions can be employed and both basis and correction functions can be updated in regions where the total mobility changes are significant. The computational cost of this algorithm can be comparable with the one of the original MSFV method. However, the accuracy of the solutions can be improved dramatically.

[00104] The i-MSFV method with infrequently updating the localization conditions is tested for two-phase flow scenarios with a viscosity ratio μ_2/μ_1 of 10. The relative permeability $k_{r_{1,2}} = S_{1,2}^2$ (where $S_{1,2} \in [0,1]$ are the phase saturations) is used for the first example and $k_{r_{1,2}} = S_{1,2}$ is used for the second example. The permeability fields of Figures 20A and 23A are employed and the rectangular domains are discretized by 220 × 55 and 55 × 55 fine grids, respectively. In both examples, coarse grids consisting of volumes containing 11 × 11 fine cells with an aspect ratio of 10 are used and no-flow conditions are applied at the whole domain boundary. Initially, the domains are saturated with a first viscous fluid and a less viscous fluid is injected into the fine cell (0, 0). In the first scenario, production occurs from cell (220, 55) and in the second scenario from cell (55, 55). For the numerical solution of the phase transport equation, an explicit scheme was employed.

Figures 24 and 25 show the saturation maps for the two test cases after 0.165 pore volume injected (PVI). The i-MSFV method that updates the correction function boundary conditions every 10th time step (shown in Figures 24B and 25B) can lead to results that are virtually identical with the fine-scale reference solutions shown in Figures 24A and 25A. The MSFV solutions that are shown in 23C and 24C of show significant deviations from the reference.

[00105] While in the foregoing specification this invention has been described in relation to certain preferred embodiments thereof, and many details have been set forth for purpose of illustration, it will be apparent to those skilled in the art that the invention is susceptible to alteration and that certain other details described herein can vary considerably without departing from the basic principles of the invention.

[00106] It is further noted that the systems and methods may be implemented on various types of computer architectures, such as for example on a single general purpose computer or workstation, or on a networked system, or in a client-server configuration, or in an application service provider configuration. An exemplary computer system suitable for implementing the methods disclosed herein is illustrated in Figure 26. As shown in Figure 26, the computer system to implement one or more methods and systems disclosed herein can be linked to a network link which can be, *e.g.*, part of a local area network to other, local computer systems and/or part of a wide area network, such as the Internet, that is connected to other, remote computer systems. For example, the methods and systems described herein may be implemented on many different types of processing devices by program code including program instructions that are executable by the device processing subsystem. The software program instructions may include source code, object code, machine code, or any

other stored data that is operable to cause a processing system to perform the methods and operations described herein. As an illustration, a computer can be programmed with instructions to perform the various steps of the flowcharts shown in Figures 4 and 5, and various steps of the processes of the block diagram shown in Figure 1.

[00107] It is further noted that the systems and methods may include data signals conveyed via networks (*e.g.*, local area network, wide area network, internet, combinations thereof), fiber optic medium, carrier waves, wireless networks, and combinations thereof for communication with one or more data processing devices. The data signals can carry any or all of the data disclosed herein that is provided to or from a device.

[00108] The systems' and methods' data (*e.g.*, associations, mappings, data input, data output, intermediate data results, final data results) may be stored and implemented in one or more different types of computer-implemented data stores, such as different types of storage devices and programming constructs (*e.g.*, RAM, ROM, Flash memory, flat files, databases, programming data structures, programming variables, IF-THEN (or similar type) statement constructs). It is noted that data structures describe formats for use in organizing and storing data in databases, programs, memory, or other computer-readable media for use by a computer program. As an illustration, a system and method can be configured with one or more data structures resident in a memory for storing data representing a fine grid, a coarse grid, a dual coarse grid, and dual basis functions calculated on the dual coarse control volumes by solving local elliptic problems. Software instructions (executing on one or more data processors) can access the data stored in the data structure for generating the results described herein).

[00109] An embodiment of the present disclosure provides a computer-readable medium storing a computer program executable by a computer for performing the steps of any of the methods disclosed herein. A computer program product can be provided for use in conjunction with a computer having one or more memory units and one or more processor units, the computer program product including a computer readable storage medium having a computer program mechanism encoded thereon, wherein the computer program mechanism can be loaded into the one or more memory units of the computer and cause the one or more processor units of the computer to execute various steps illustrated in the flow chart of Figures 4 and 5, and various steps of the processes of the block diagram shown in Figure 1.

[00110] The computer components, software modules, functions, data stores and data structures described herein may be connected directly or indirectly to each other in order to allow the flow of data needed for their operations. It is also noted that a module or processor includes but is not limited to a unit of code that performs a software operation, and can be implemented for example as a subroutine unit of code, or as a software function unit of code, or as an object (as in an object-oriented paradigm), or as an applet, or in a computer script language, or as another type of computer code. The software components and/or functionality may be located on a single computer or distributed across multiple computers depending upon the situation at hand.

[00111] It should be understood that as used in the description herein and throughout the claims that follow, the meaning of "a," "an," and "the" includes plural reference unless the context clearly dictates otherwise. Also, as used in the description herein and throughout the claims that follow, the meaning of "in" includes "in" and "on" unless the context clearly dictates otherwise. Finally, as used in the description herein and throughout the claims that

follow, the meanings of "and" and "or" include both the conjunctive and disjunctive and may be used interchangeably unless the context expressly dictates otherwise.

WHAT IS CLAIMED IS:

1. A multi-scale computer-implemented method for use in modeling fluid flow in a subsurface reservoir, the method comprising:

creating a fine grid defining a plurality of fine cells associated with a geological formation of the subsurface reservoir, a coarse grid defining a plurality of coarse cells having interfaces between the coarse cells, the coarse cells being aggregates of the fine cells, and a dual coarse grid defining a plurality of dual coarse control volumes, the dual coarse control volumes being aggregates of the fine cells and having boundaries bounding the dual coarse control volumes;

calculating basis functions on the dual coarse control volumes by solving local elliptic problems;

integrating a source term of an elliptic pressure equation over each coarse cell; and

calculating a pressure using an iterative method comprising, for each iteration:

(i) applying a smoothing scheme to a solution for the pressure over the fine grid from a previous iteration to provide a smoothed fine-grid pressure;

(ii) calculating correction functions using the smoothed fine-grid pressure from step (i);

(iii) applying a restriction operation comprising the correction functions from step (ii) to solve for the pressure over the coarse grid; and

(iv) applying a prolongation operation to the pressure solved over the coarse grid from step (iii) to reconstruct an updated solution for the pressure over the fine-grid;

wherein the pressure calculated using the iterative method is used to model fluid flow in the subsurface reservoir.

2. The method of claim 1, wherein steps (i) to (iv) are repeated until the solution for the pressure over the fine-grid converges.

3. The method of claim 1, further comprising outputting or displaying the pressure calculated using the iterative method.

4. The method of claim 1, further comprising, prior to calculating the pressure using the iterative method, a step of initializing the value for the pressure in the fine cells by setting the value equal to zero.

5. The method of claim 1, wherein the solution for the pressure over the fine grid in step (i) is calculated using the calculated basis functions and the integrated source term.

6. The method of claim 1, wherein the iterative method is performed for a plurality of timesteps and the dual basis functions and the correction functions are re-computed in at least one timestep over the dual coarse control volumes where a change of the mobility coefficient of the local elliptic problems exceeds a predetermined threshold value.

7. The method of claim 1, wherein the smoothing scheme of step (i) comprises applying n_s smoothing steps, where n_s is a positive integer greater than 1.

8. The method of claim 1, wherein the step of applying the smoothing scheme to the solution of the pressure over the fine grid comprises applying a line-relaxation smoothing operation, the line-relaxation smoothing operation comprising:

applying a linear operator that has a tri-diagonal structure to the solution of the pressure over the fine grid to provide a linear system of equations; and

solving the linear system of equations using a Thomas algorithm.

9. The method of claim 1, wherein the prolongation operation comprises the correction functions superimposed with basis functions that have been weighted with the values for the pressure obtained from step (iii).

10. The method of claim 1, wherein the calculating the pressure using the iterative method is performed for at least one timestep in a plurality of timesteps.

11. The method of claim 1, wherein the iterative method is performed for a plurality of timesteps and the correction functions are re-computed in at least one timestep where the source term exceeds a predetermined limit.

12. The method of claim 1, wherein the applying the restriction operation comprising the correction functions from step (ii) to solve for the pressure over the coarse grid of step (iii) comprises:

calculating the right-hand side of the linear system for the pressure over the coarse-grid using the correction functions from step (ii); and

solving for the pressure on the coarse grid using the calculated right-hand side of the linear system for the pressure over the coarse-grid.

13. A computer-implemented system for use in modeling fluid flow in a geological formation of a subsurface reservoir using a model, the system comprising:

one or more data structures resident in a memory for storing data representing a fine grid, a coarse grid, a dual coarse grid, and dual basis functions calculated on the dual coarse control volumes by solving local elliptic problems; and

software instructions, for executing on one or more data processors, to compute the model using a finite volume method in at least two timesteps; wherein:

the model comprises one or more variables representative of fluid flow in the subsurface reservoir, wherein at least one of the one or more variables representative of fluid flow is responsive to calculated basis functions;

the computing comprises:

calculating the basis functions on the dual coarse control volumes by solving local elliptic problems;

integrating a source term of an elliptic pressure equation over each coarse cell; and

for at least one timestep of the at least two timesteps, calculating a pressure using an iterative method, the iterative method comprising, for each iteration:

(i) applying a smoothing scheme to a solution for the pressure over the fine-grid from a previous iteration to provide a smoothed fine-grid pressure;

(ii) calculating correction functions using the smoothed fine-grid pressure from step (i);

(iii) solving for the pressure on the coarse grid using the correction functions from step (ii); and

(iv) reconstructing a solution for the pressure over the fine-grid using the result from step (iii); and

a visual display for displaying fluid flow in the geological formation of the subsurface reservoir using the computed model, comprising the pressure calculated using the iterative method in the at least two timesteps.

14. A method for operating a subsurface reservoir to achieve improved production of a reservoir fluid from a geological formation of the subsurface reservoir, comprising:

injecting a displacement fluid into a portion of the geological formation of the subsurface reservoir; and

applying a reservoir fluid production process to the subsurface reservoir under at least one operational condition that is derived based on the pressure calculated using the iterative method from any one of claims 1 and 13.

15. The method of claim 14, wherein the at least one operational condition is displacement fluid injection rate, reservoir fluid production rate, viscosity ratio of displacement fluid to reservoir fluid, location of injection of the displacement fluid, location of production of the reservoir fluid, displacement fluid saturations, reservoir fluid saturations, displacement fluid saturations at different pore volumes injected, or reservoir fluid saturations at different pore volumes injected.

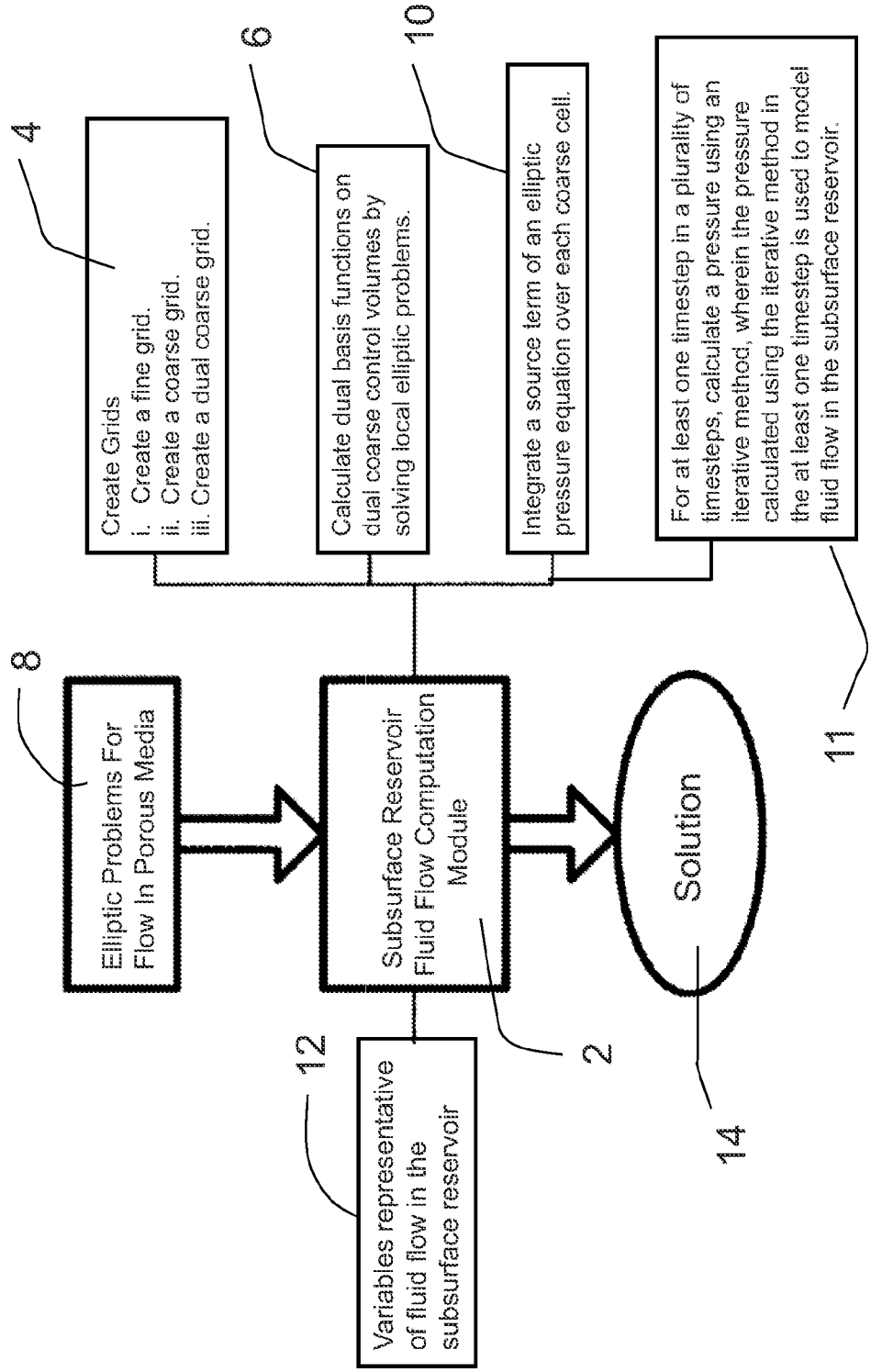

Create Grids
i. Create a fine grid.
ii. Create a coarse grid.
iii. Create a dual coarse grid.
4
Calculate dual basis functions on dual coarse control volumes by solving local elliptic problems.
6
Integrate a source term of an elliptic pressure equation over each coarse cell.
10
For at least one timestep in a plurality of timesteps, calculate a pressure using an iterative method, wherein the pressure calculated using the iterative method in the at least one timestep is used to model fluid flow in the subsurface reservoir.
11
Elliptic Problems For Flow In Porous Media
8
Subsurface Reservoir Fluid Flow Computation Module
2
Solution
14
Variables representative of fluid flow in the subsurface reservoir
12

FIG. 1

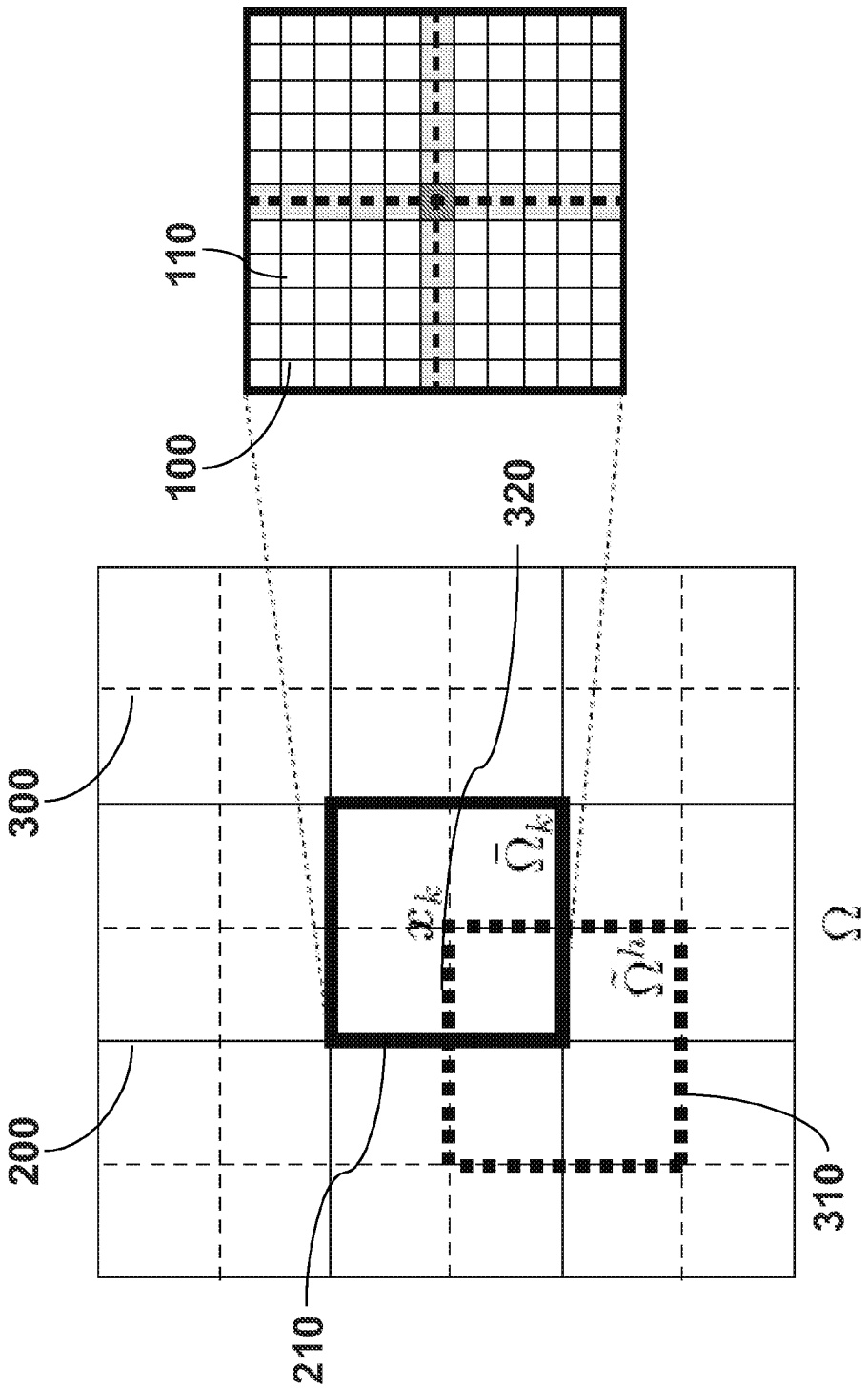
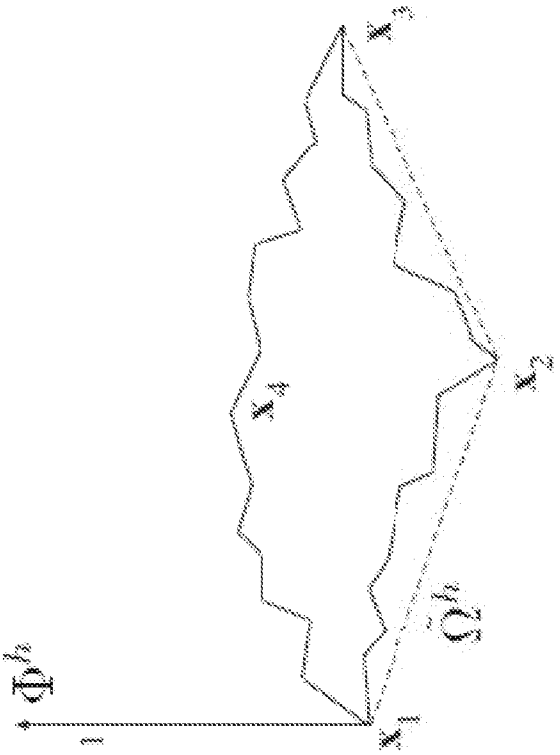
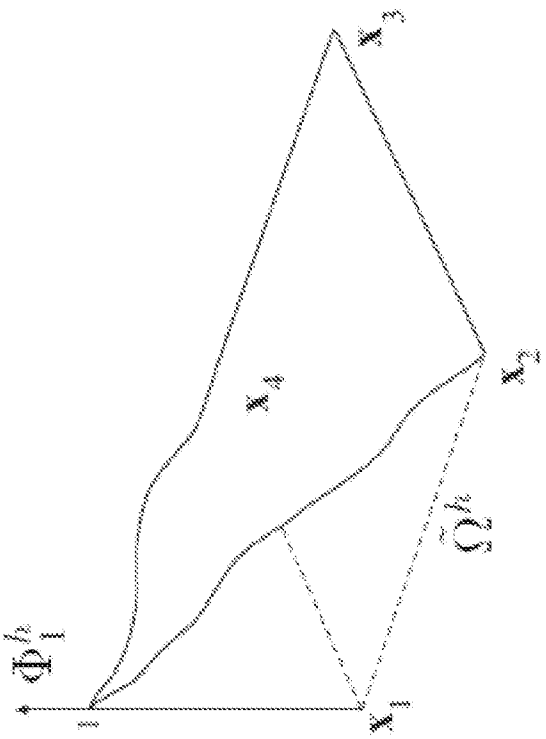


FIG. 2



(B)

FIG. 3B



(A)

FIG. 3A

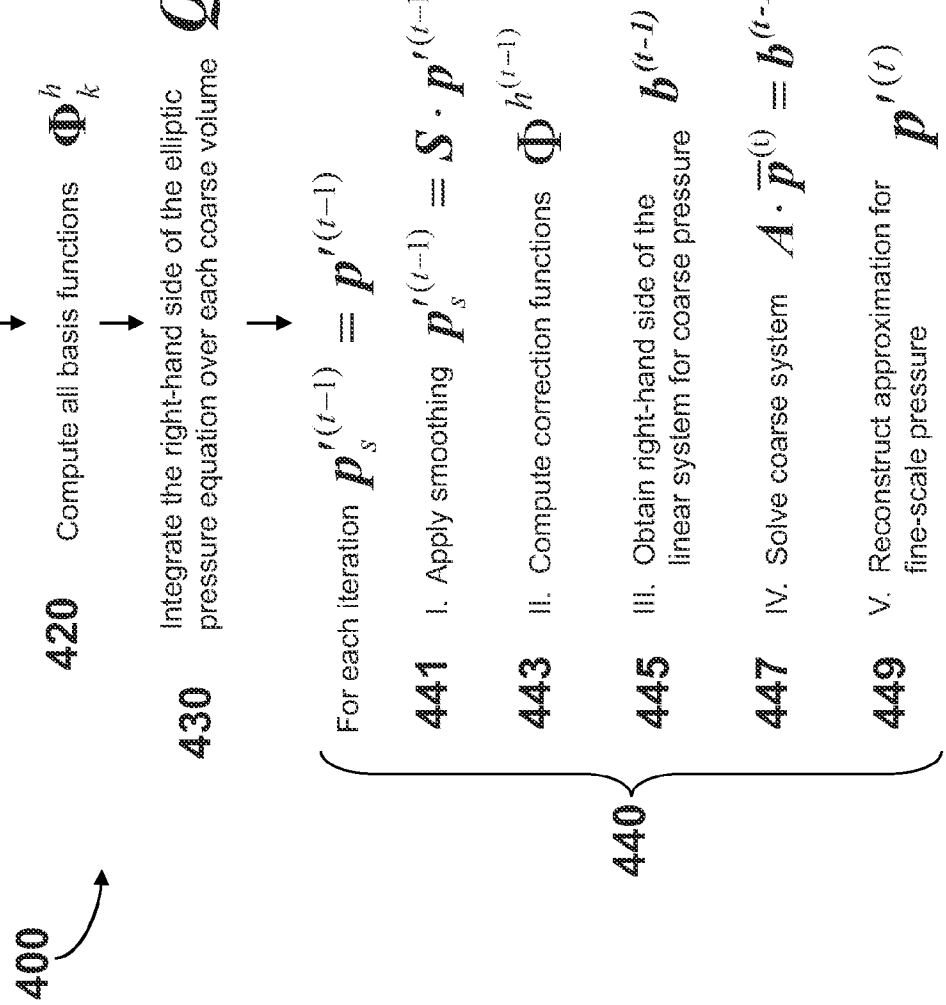

400
410 Initialize fine-scale pressure $p'^{(t=0)}$
420 Compute all basis functions Φ_k^h
430 Integrate the right-hand side of the elliptic pressure equation over each coarse volume Q
440
For each iteration $p_s'^{(t-1)} = p'^{(t-1)}$
441 I. Apply smoothing $p_s'^{(t-1)} = S \cdot p'^{(t-1)}$
443 II. Compute correction functions $\Phi^{h(t-1)}$
445 III. Obtain right-hand side of the linear system for coarse pressure $b^{(t-1)} = Q + DC \cdot p_s'^{(t-1)} + DE$
447 IV. Solve coarse system $A \cdot \bar{p}^{(t)} = b^{(t-1)}$
449 V. Reconstruct approximation for fine-scale pressure $p'^{(t)}$

FIG. 4

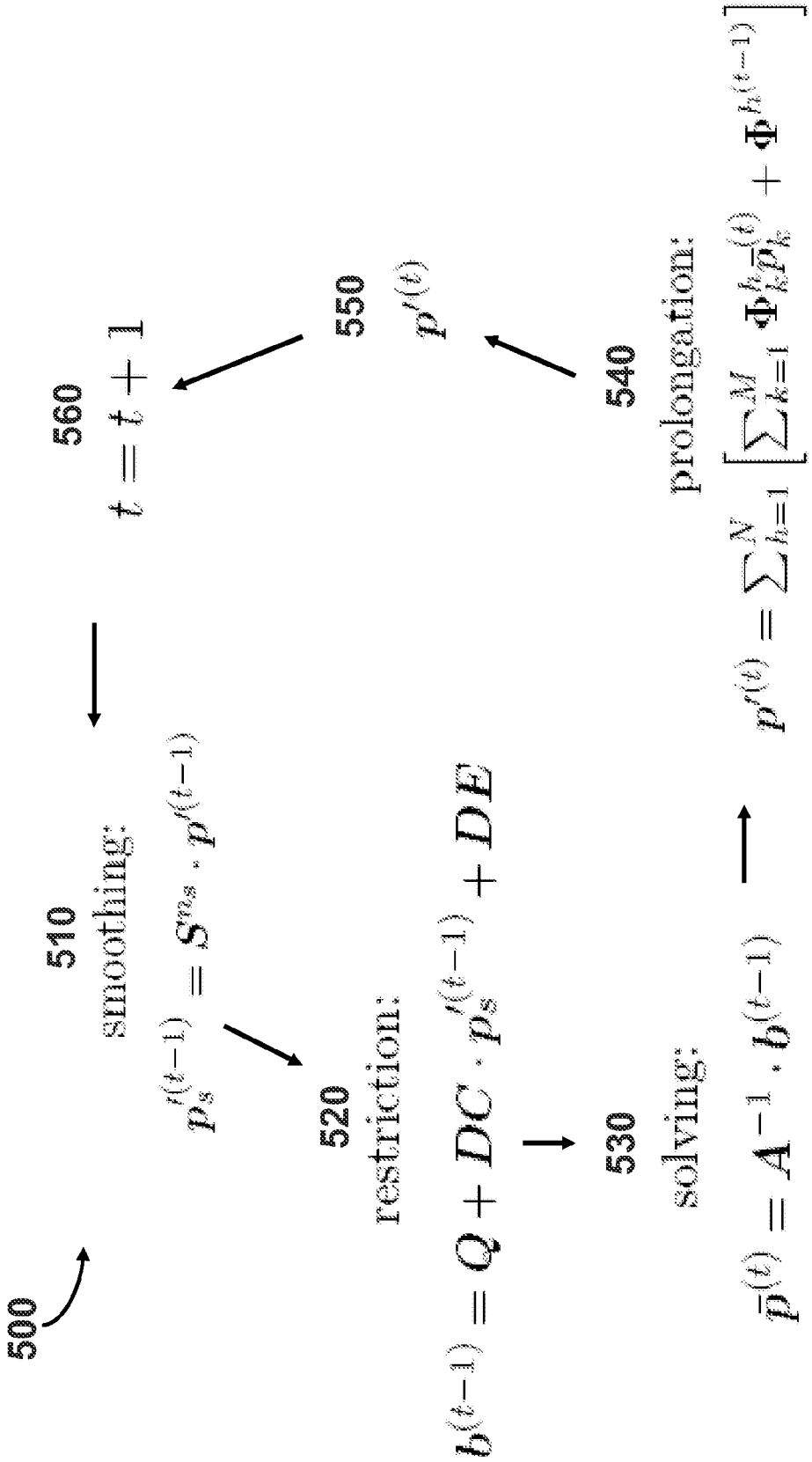

500
510
smoothing:
$p_s^{\prime(t-1)} = S^{n_s} \cdot p^{\prime(t-1)}$
520
restriction:
$b^{(t-1)} = Q + DC \cdot p_s^{\prime(t-1)} + DE$
530
solving:
$\bar{p}^{(t)} = A^{-1} \cdot b^{(t-1)}$
540
prolongation:
$p^{\prime(t)} = \sum_{h=1}^{N} \left[\sum_{k=1}^{M} \Phi_k^h \bar{p}_k^{(t)} + \Phi^{h^{(t-1)}} \right]$
550
$p^{\prime(t)}$
560
$t = t + 1$

FIG. 5

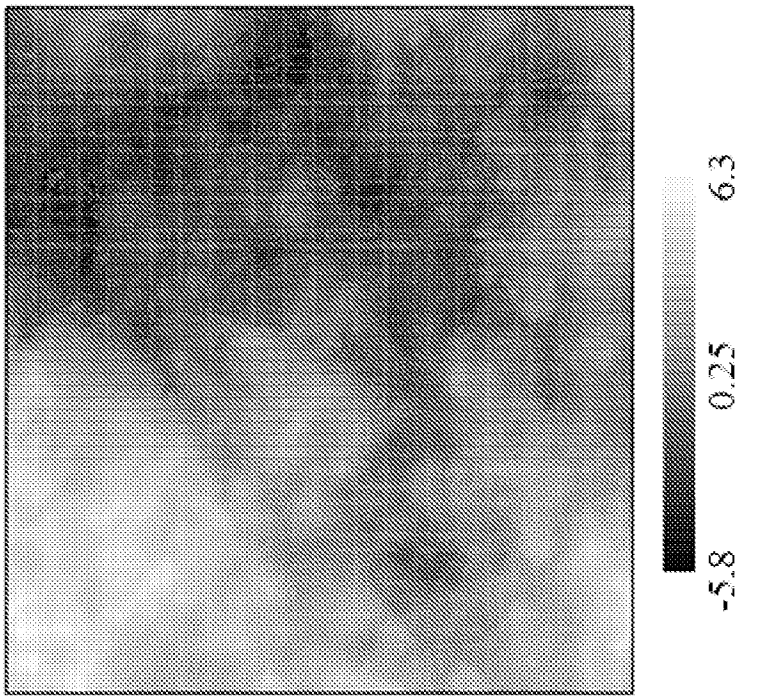


FIG. 6B

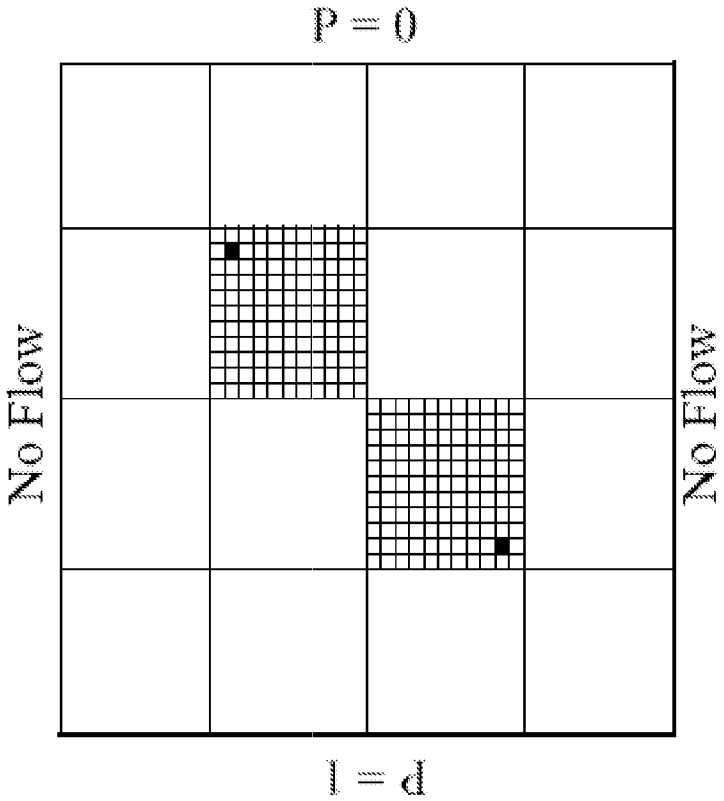


FIG. 6A

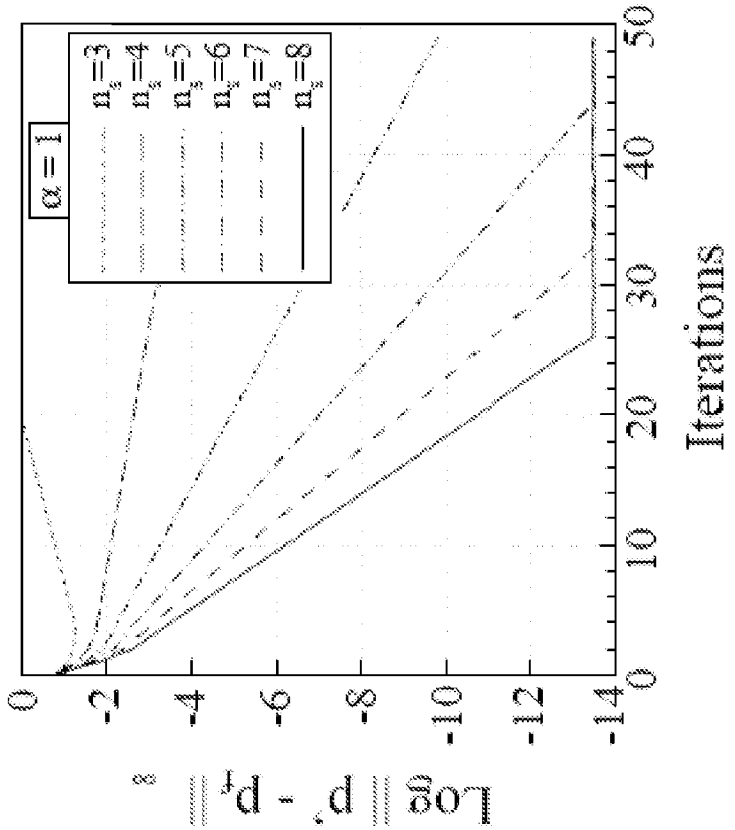

α = 1
n_s=3
n_s=4
n_s=5
n_s=6
n_s=7
n_s=8
0
-2
-4
-6
-8
-10
-12
-14
0
10
20
30
40
50
Iterations
Log || p' - p_r ||_∞

FIG. 7B

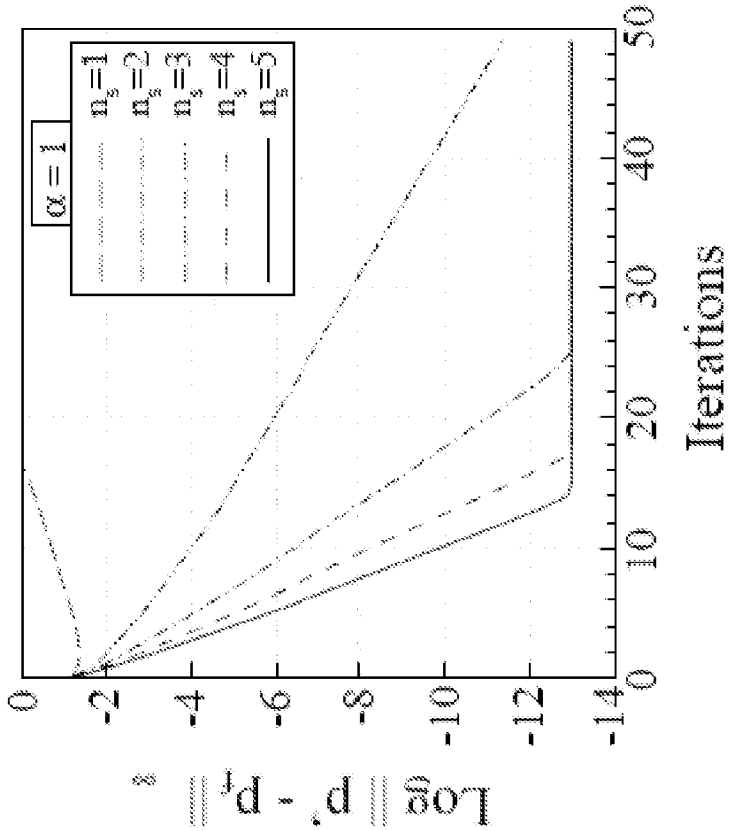

α = 1
n_s=1
n_s=2
n_s=3
n_s=4
n_s=5
0
-2
-4
-6
-8
-10
-12
-14
0
10
20
30
40
50
Iterations
Log || p' - p_r ||_∞

FIG. 7A

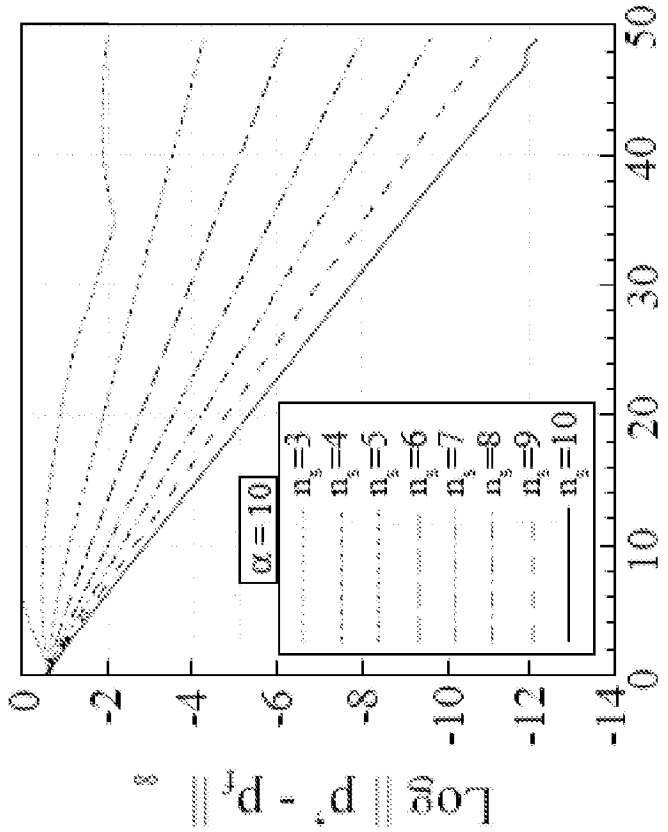

Log || p' - p_r ||_∞
Iterations
α = 10
n_s=3
n_s=4
n_s=5
n_s=6
n_s=7
n_s=8
n_s=9
n_s=10

FIG. 7D

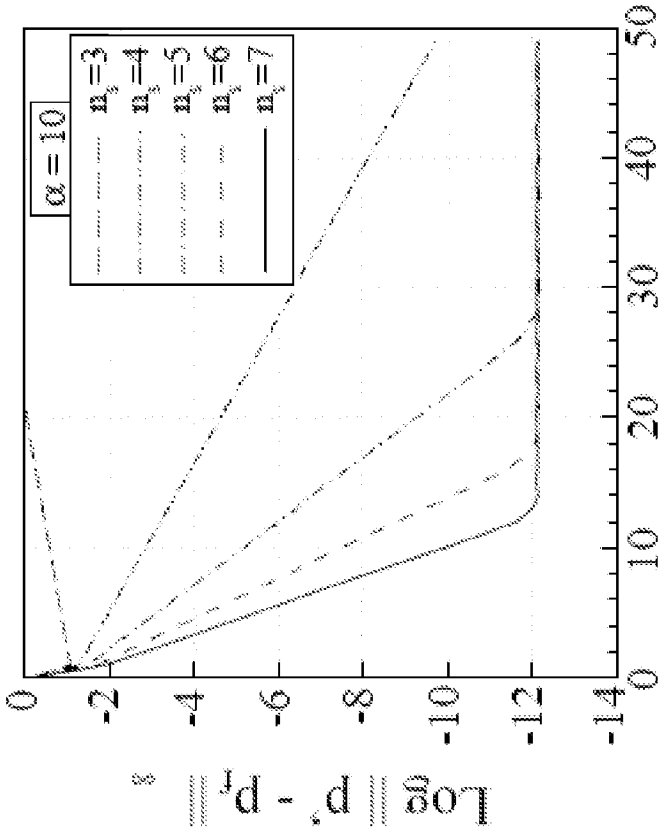

Log || p' - p_r ||_∞
Iterations
α = 10
n_s=3
n_s=4
n_s=5
n_s=6
n_s=7

FIG. 7C

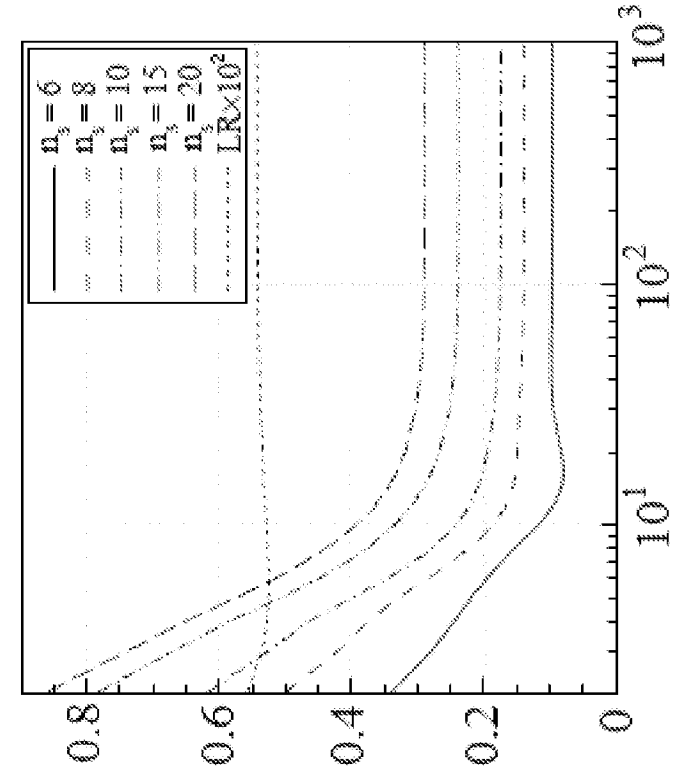

Convergence Rate
0.8
0.6
0.4
0.2
0
10^1
10^2
10^3
α
$n_s = 6$
$n_s = 8$
$n_s = 10$
$n_s = 15$
$n_s = 20$
LR×10^2

FIG. 8B

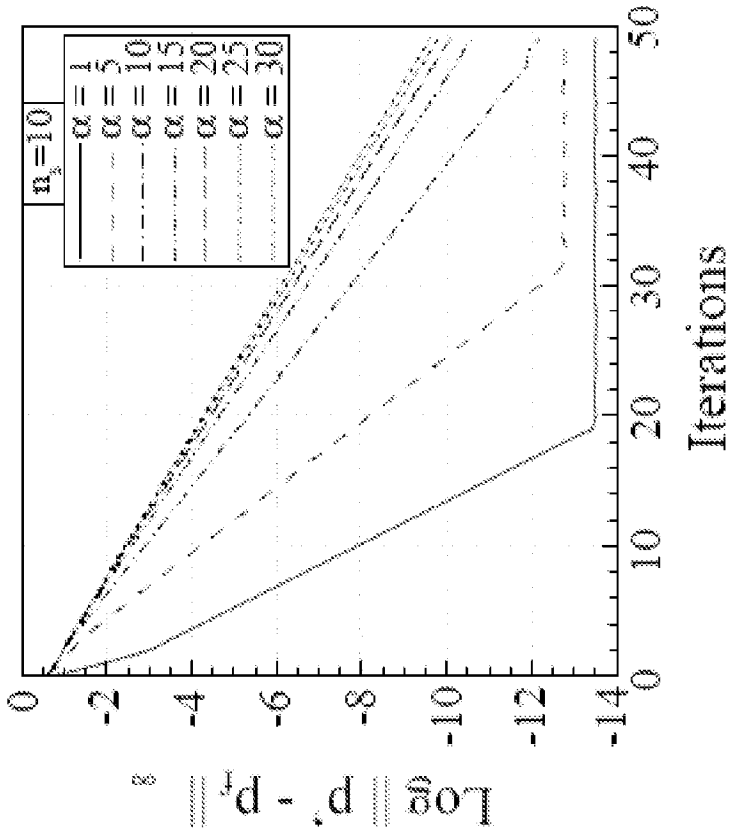

Log || p' - p_r ||_∞
0
-2
-4
-6
-8
-10
-12
-14
0
10
20
30
40
50
Iterations
n_s=10
α = 1
α = 5
α = 10
α = 15
α = 20
α = 25
α = 30

FIG. 8A

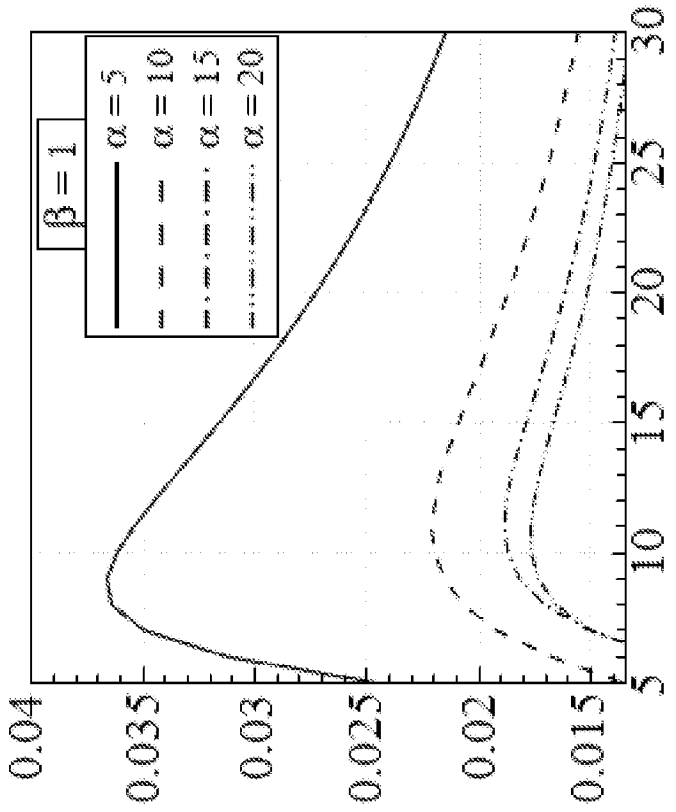

Effective Convergence Rate
0.04
0.035
0.03
0.025
0.02
0.015
5
10
15
20
25
30
n_s
β = 1
α = 5
α = 10
α = 15
α = 20

FIG. 9B

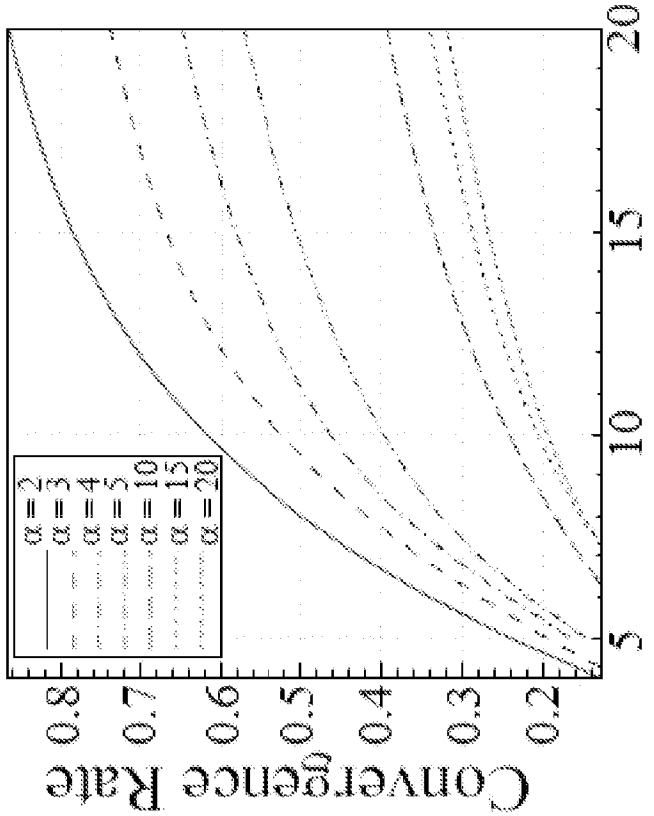

Convergence Rate
0.8
0.7
0.6
0.5
0.4
0.3
0.2
5
10
15
20
n_s
α = 2
α = 3
α = 4
α = 5
α = 10
α = 15
α = 20

FIG. 9A

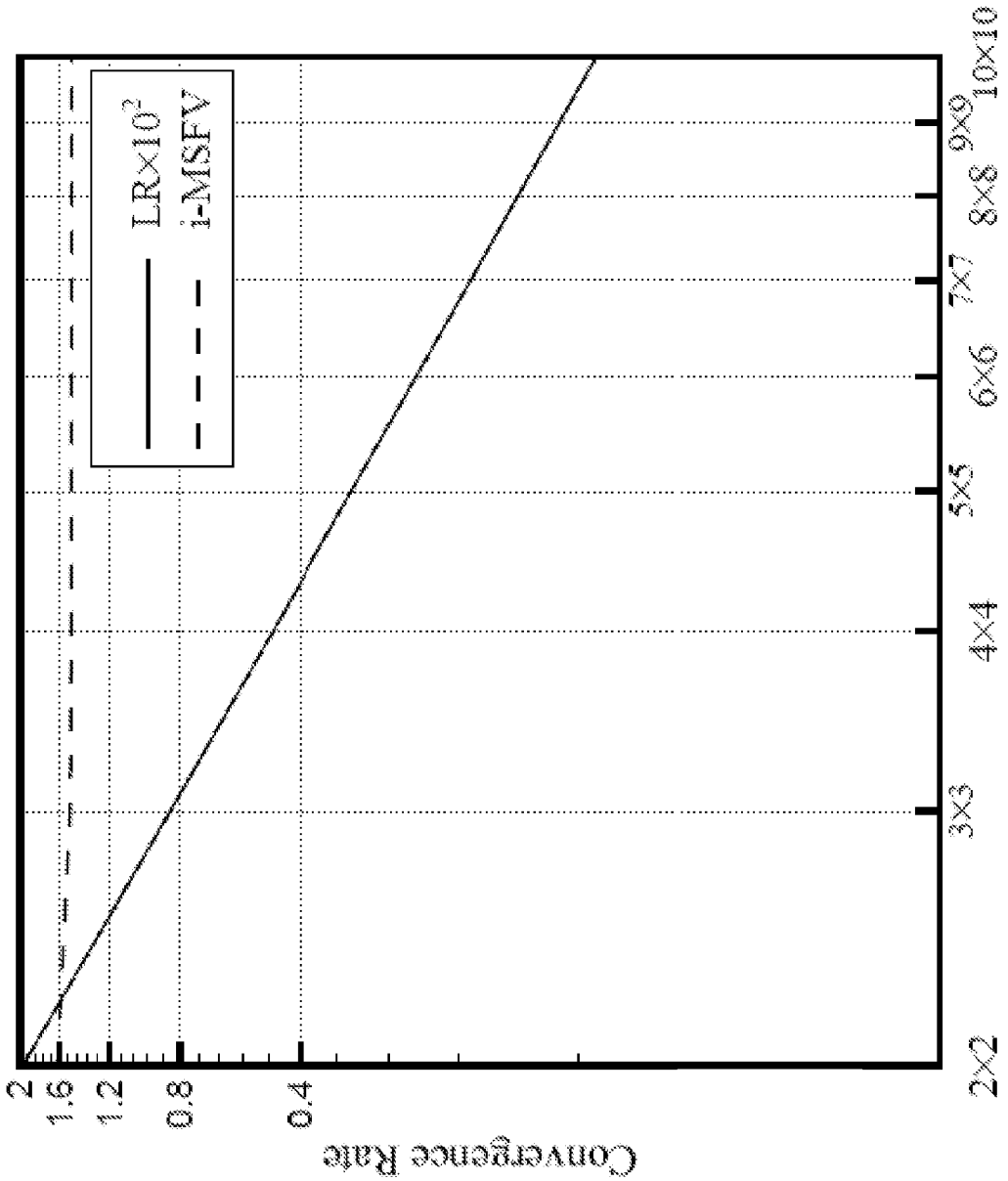

Convergence Rate
2
1.6
1.2
0.8
0.4
LR×10^2
i-MSFV
2X2
3X3
4X4
5X5
6X6
7X7
8X8
9X9
10X10
Size of Coarse Grid

FIG. 10

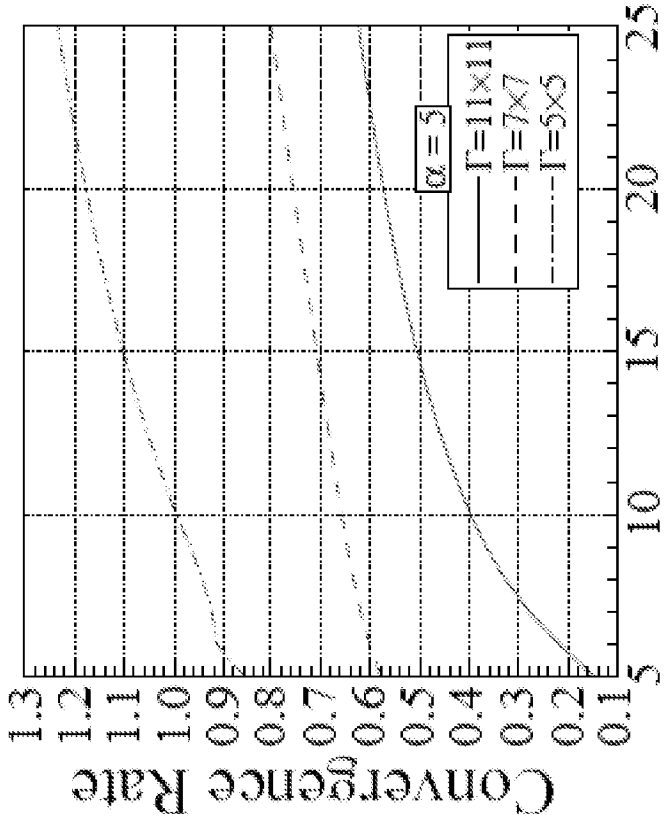


FIG. 11B

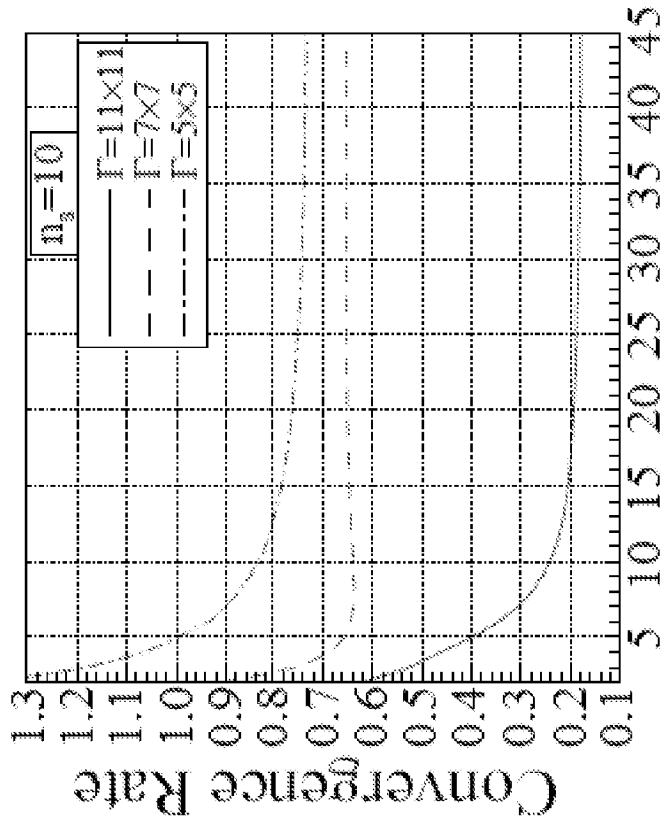


FIG. 11A

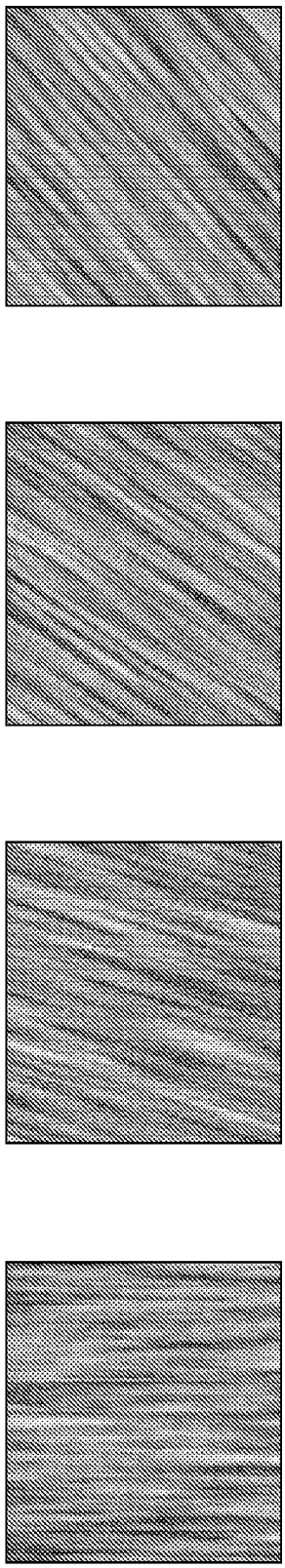


FIG. 12A FIG. 12B FIG. 12C FIG. 12D

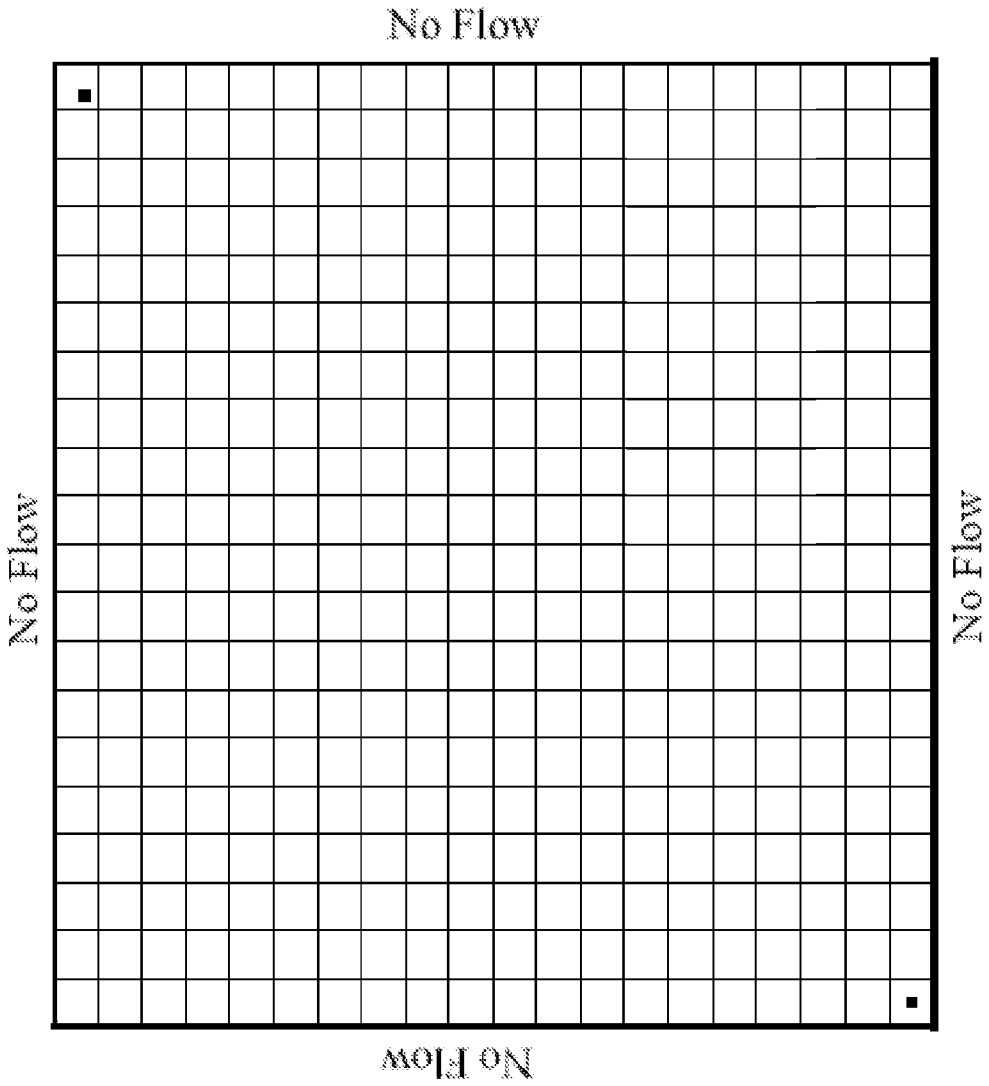


FIG. 13

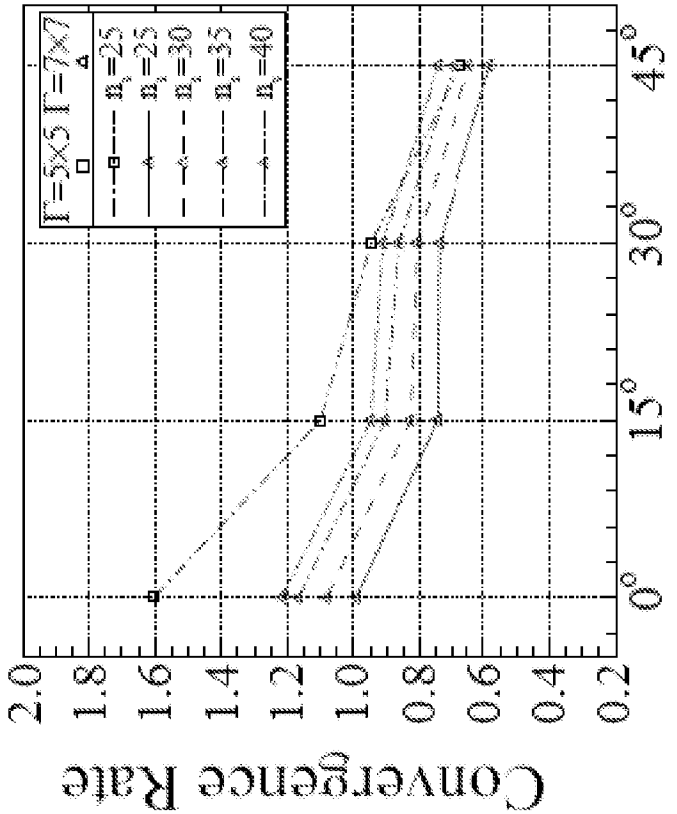


FIG. 14B

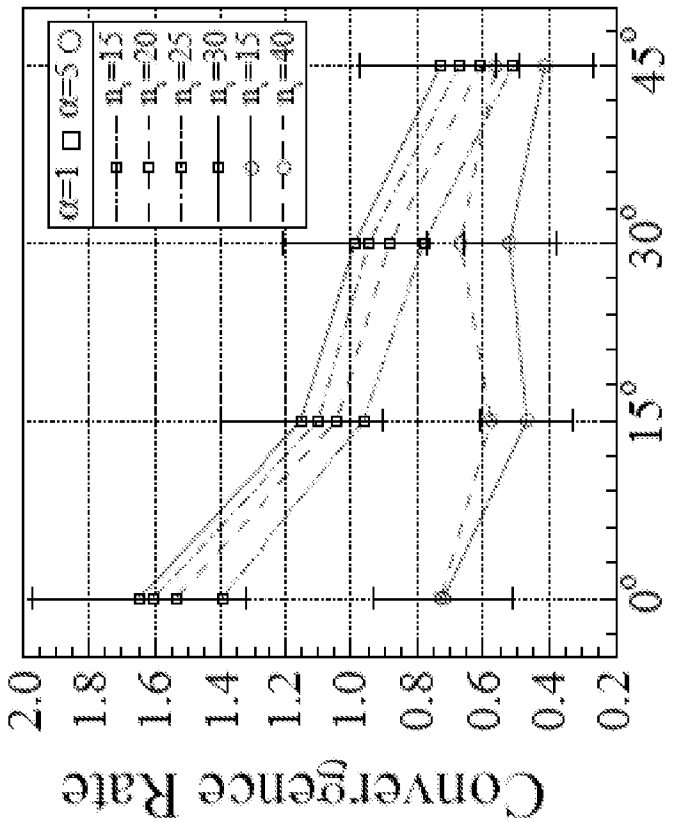


FIG. 14A

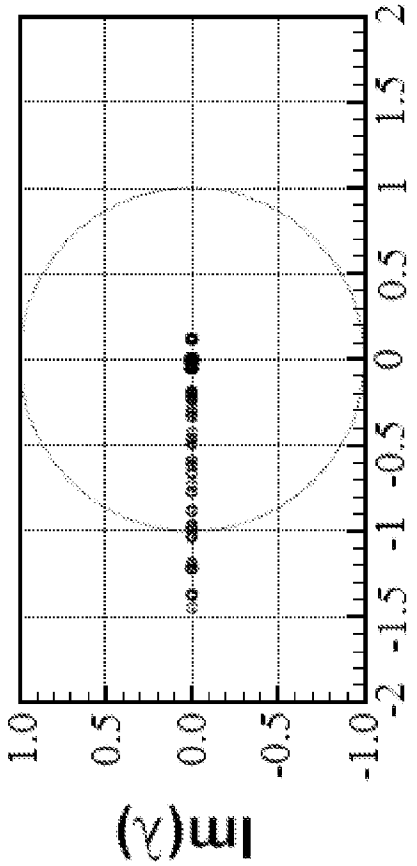


FIG. 15B

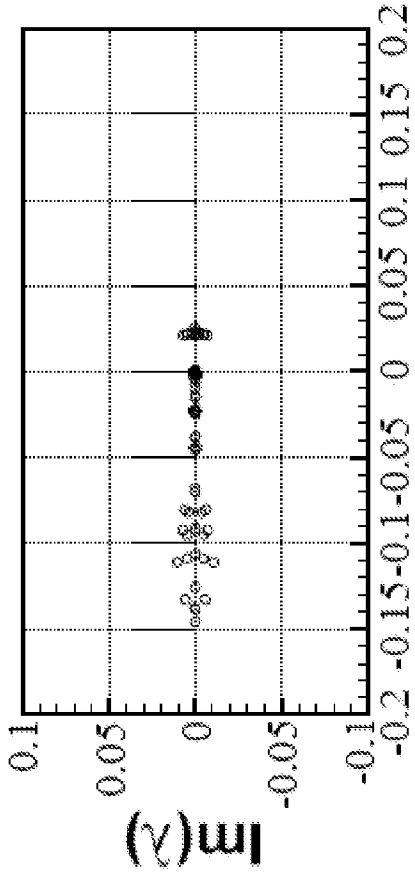


FIG. 15D

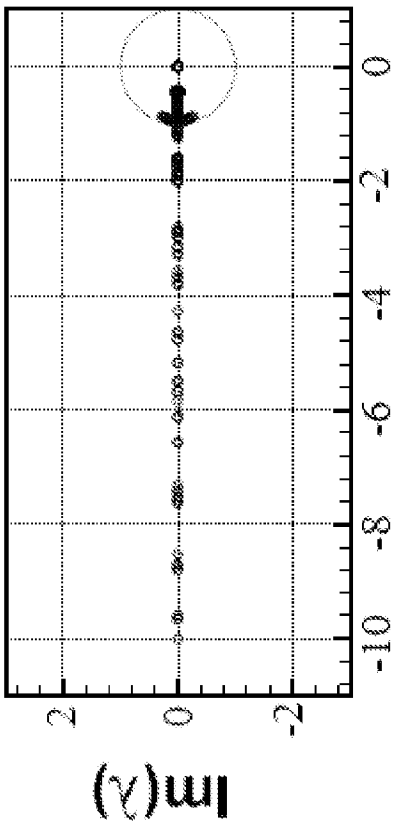


FIG. 15A

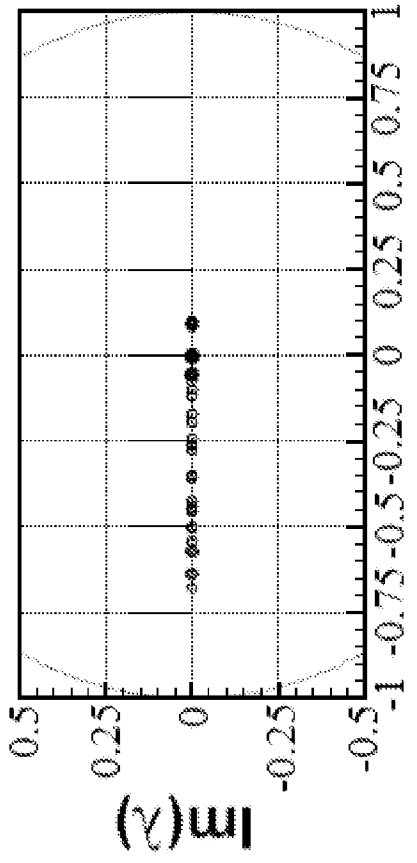


FIG. 15C

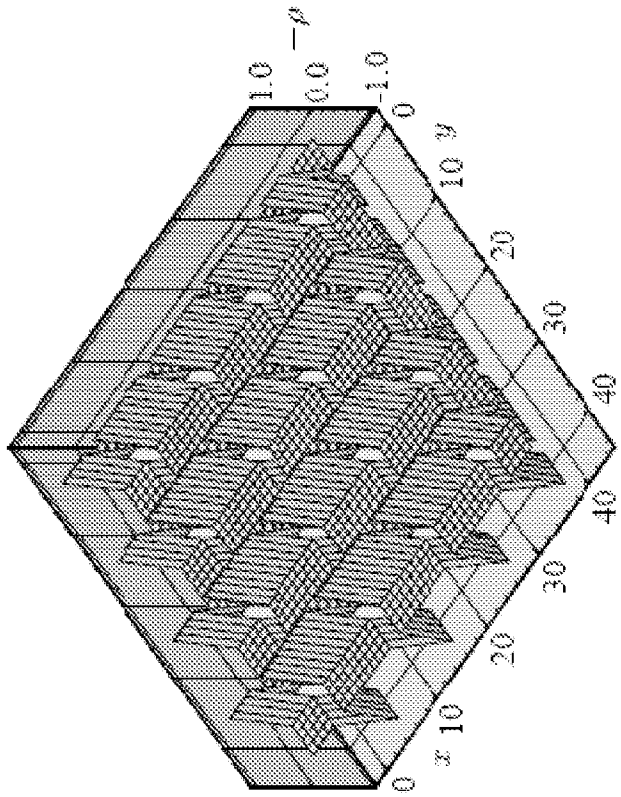


FIG. 16B

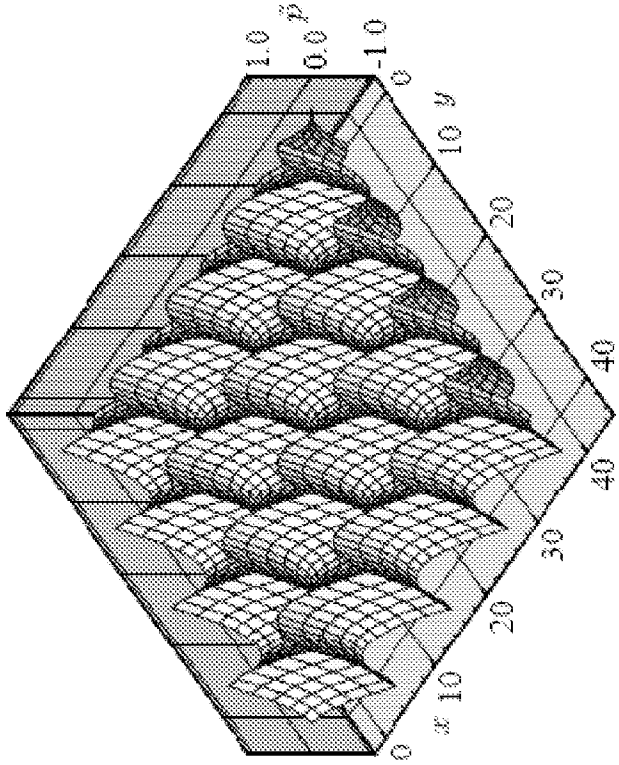


FIG. 16A

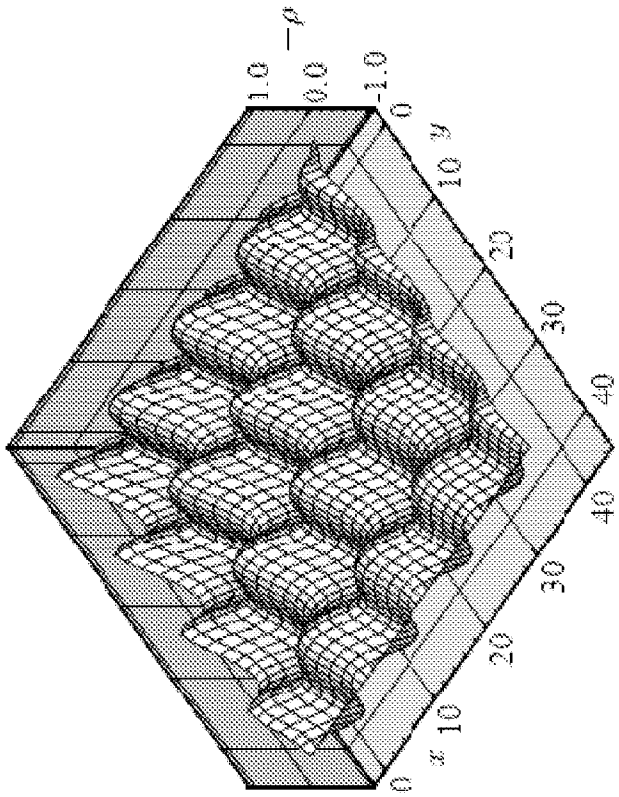


FIG. 16D

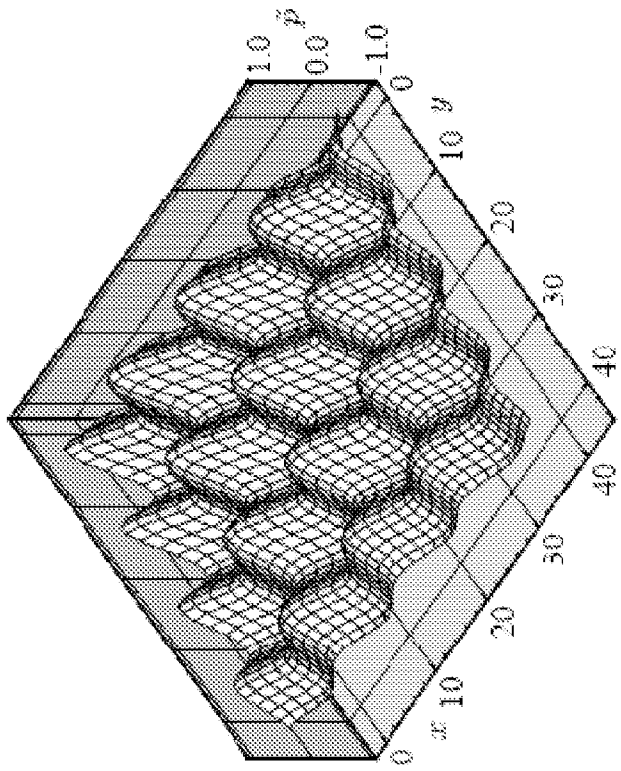


FIG. 16C

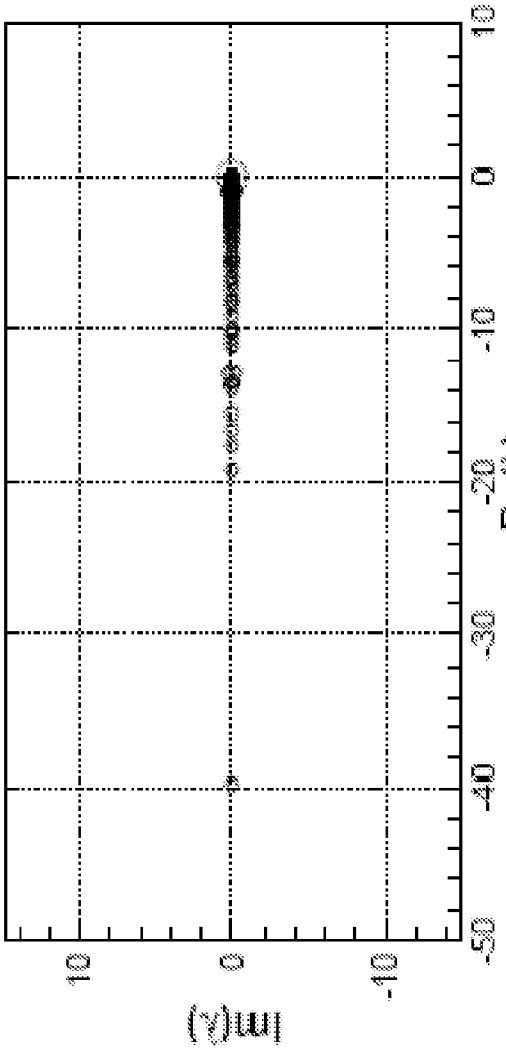


FIG. 17A

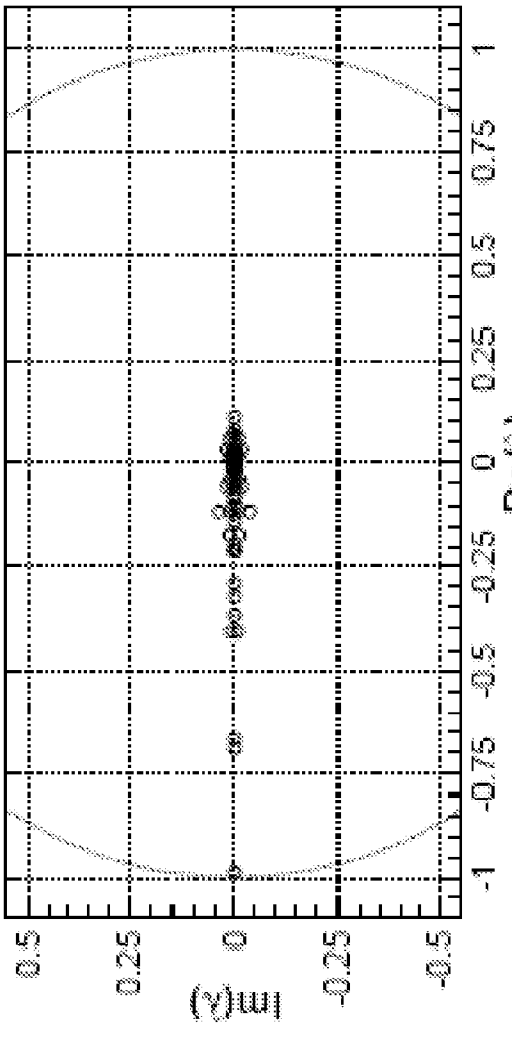


FIG. 17B

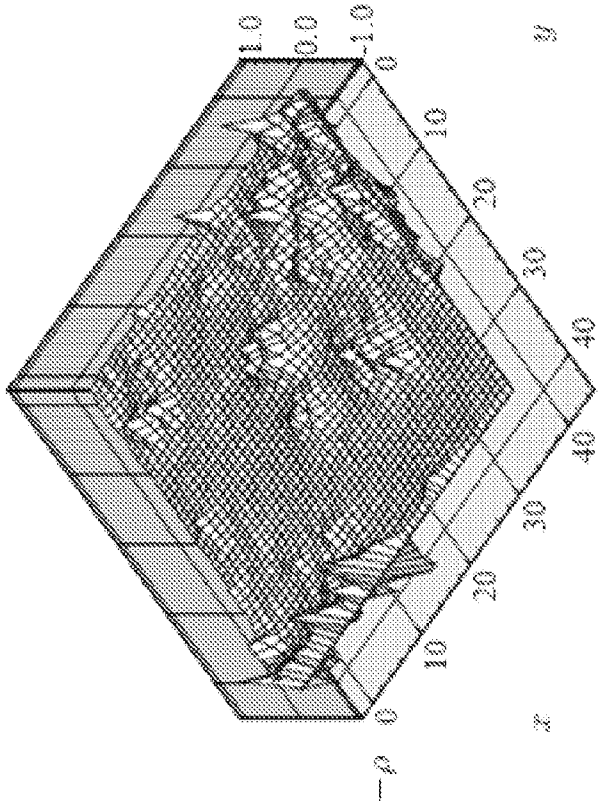


FIG. 18B

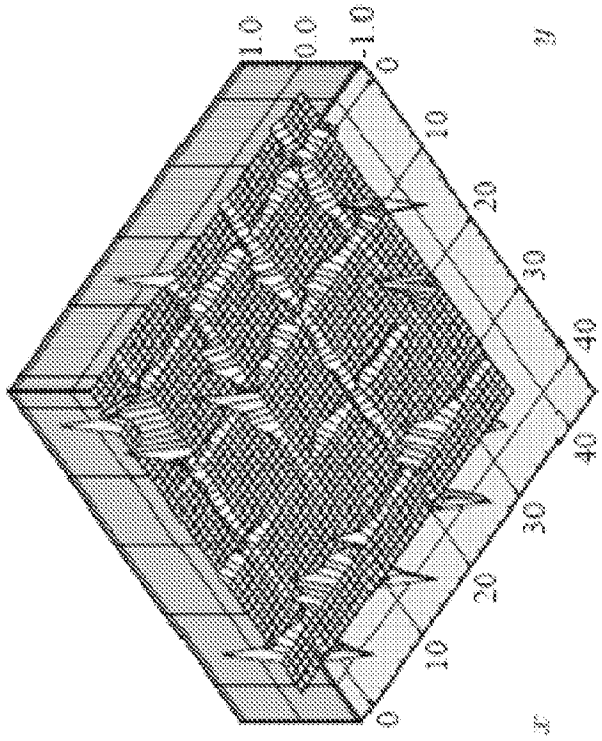


FIG. 18A

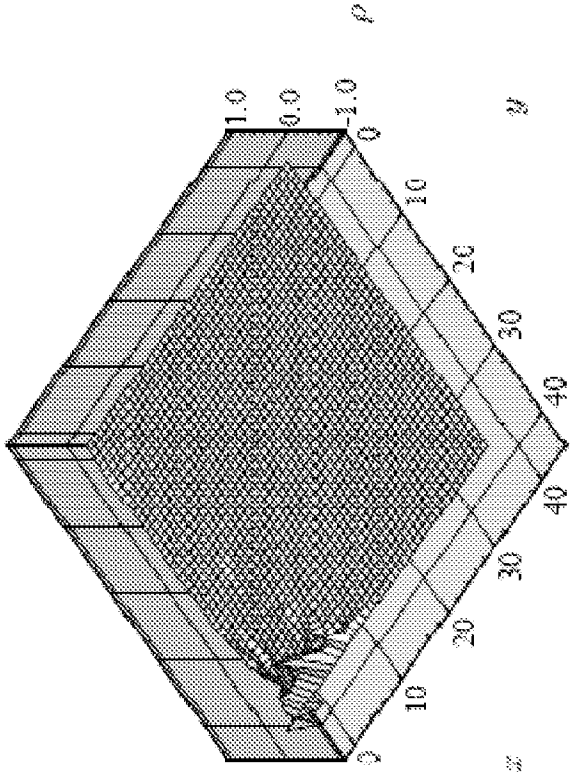


FIG. 19B

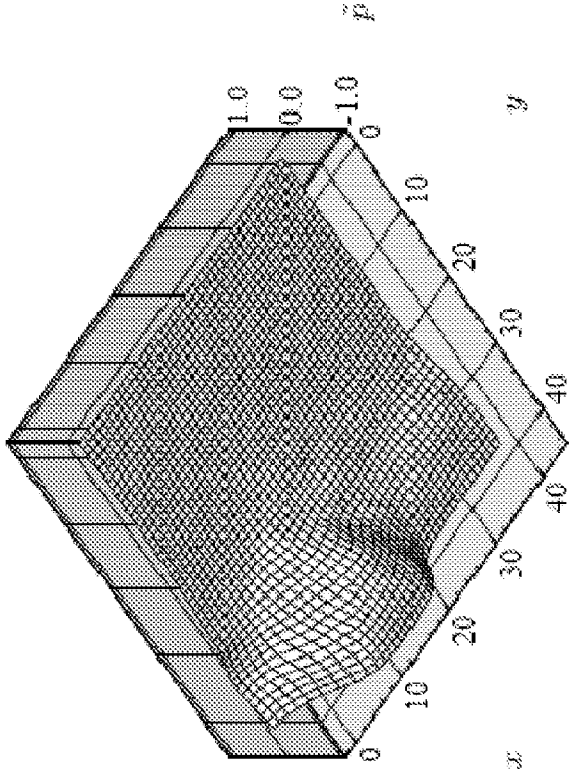


FIG. 19A

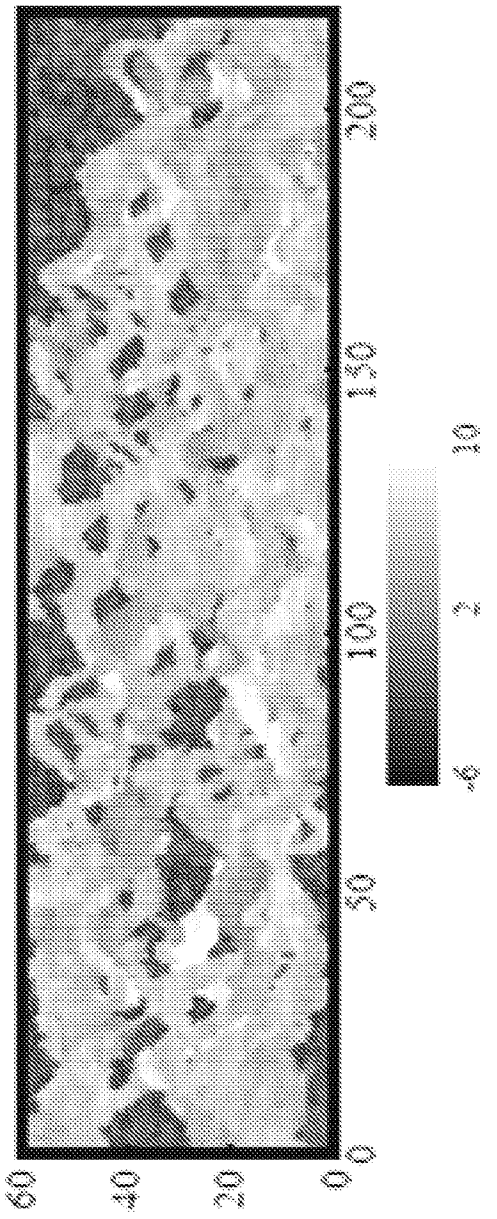


FIG. 20A

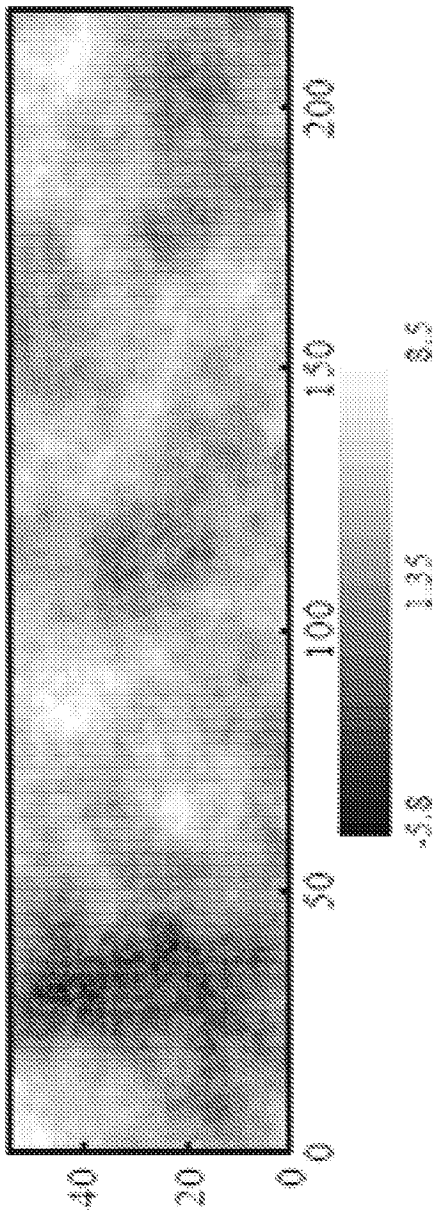


FIG. 20B

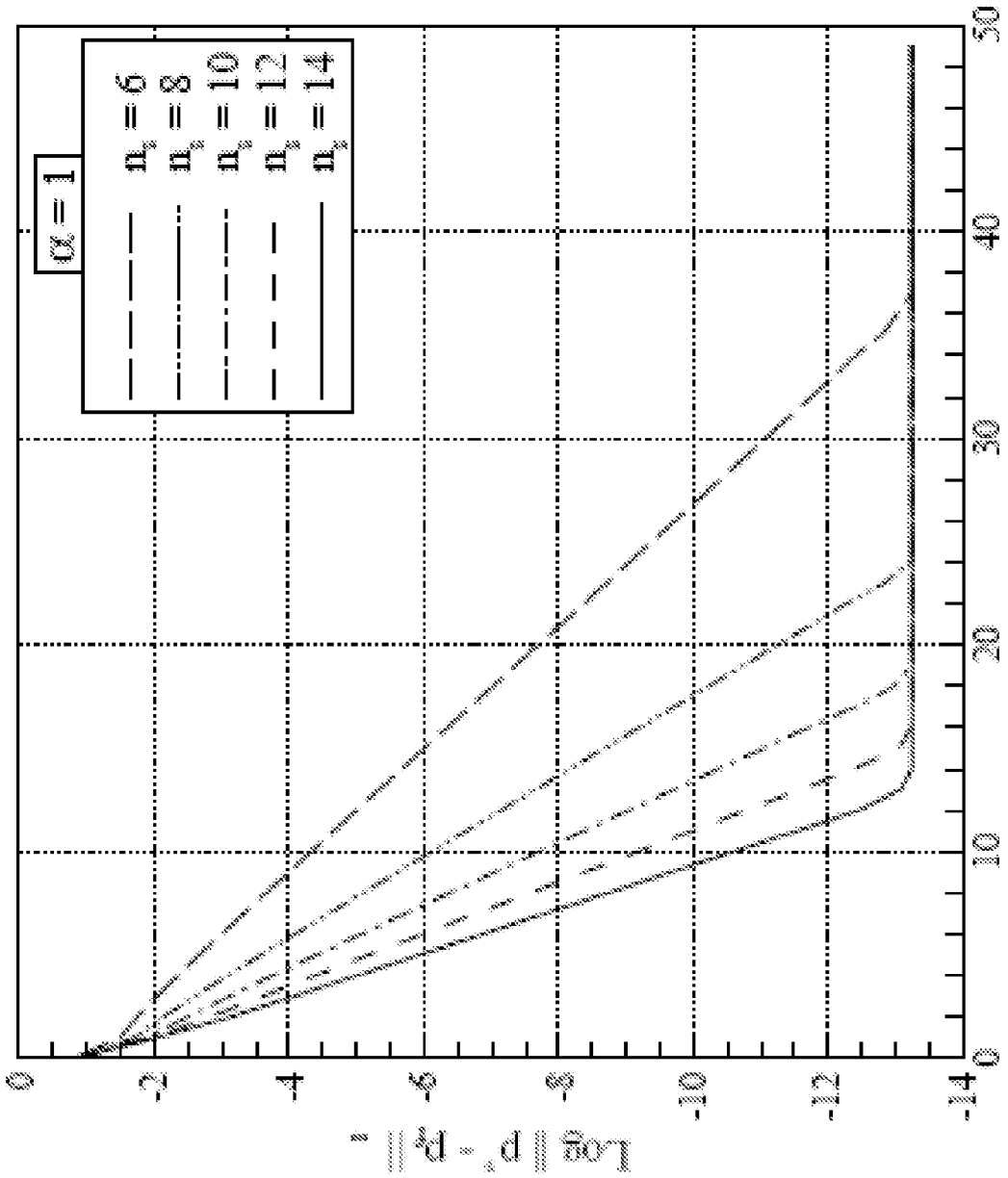

α = 1
n_s = 6
n_s = 8
n_s = 10
n_s = 12
n_s = 14
Iterations
0
10
20
30
40
50
0
-2
-4
-6
-8
-10
-12
-14

FIG. 21A

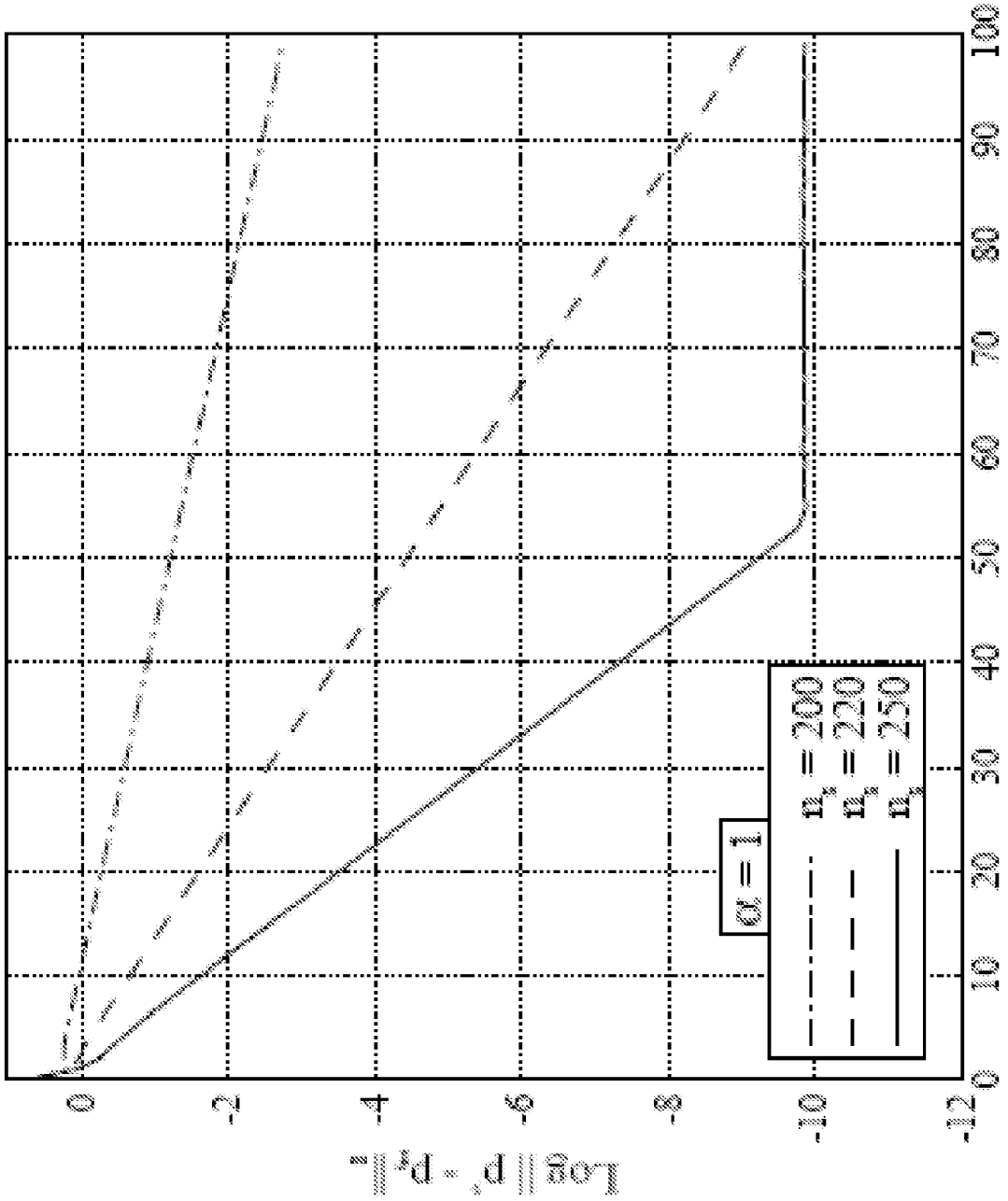

Log || p^r - p_r ||
Iterations
0
-2
-4
-6
-8
-10
-12
10
20
30
40
50
60
70
80
90
100
α = 1
n_r = 200
n_r = 220
n_r = 250

FIG. 21B

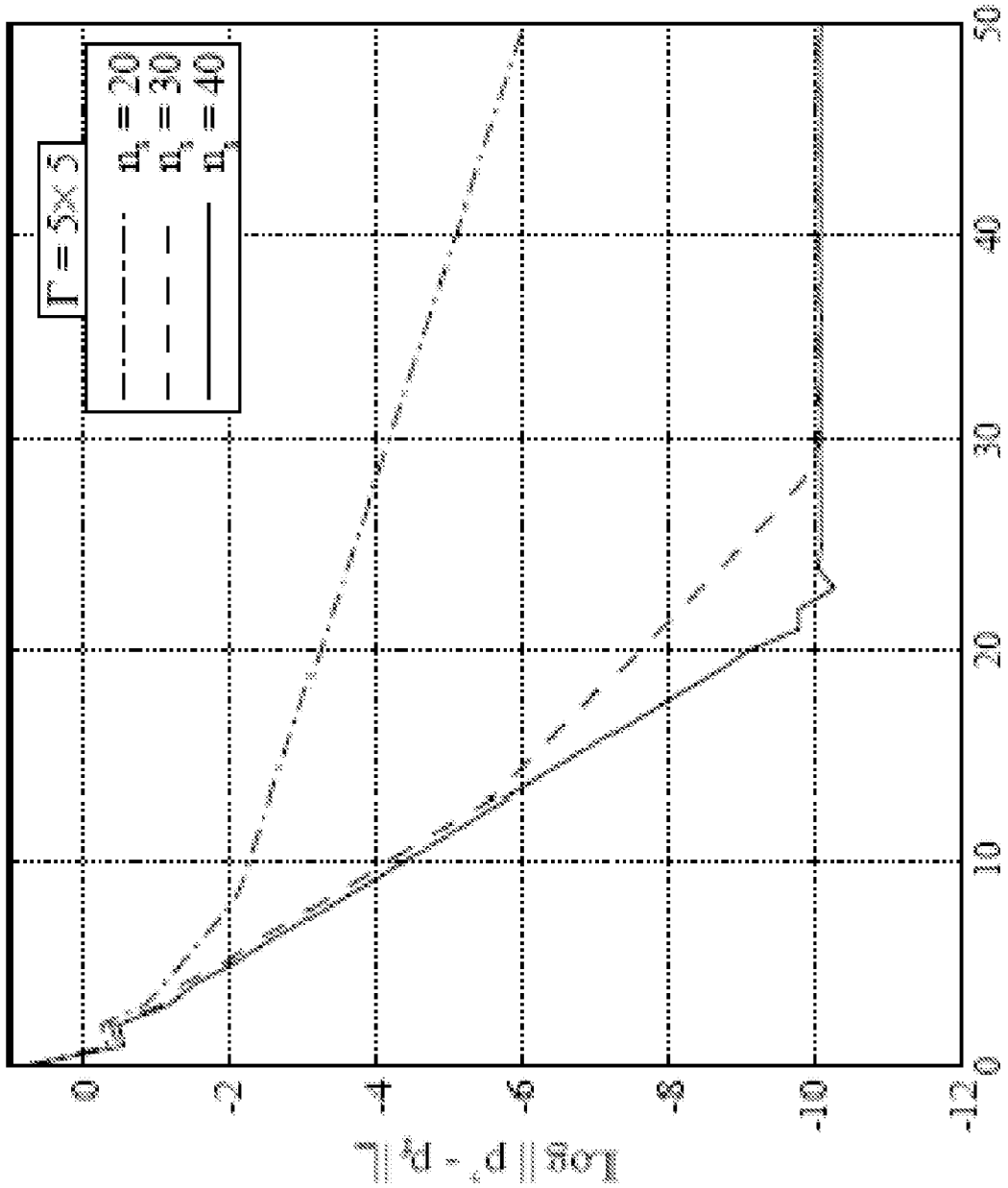

Γ = 5x5
n1 = 20
n1 = 30
n1 = 40
Log || P' - P ||
Iterations
0
-2
-4
-6
-8
-10
-12
0
10
20
30
40
50

FIG. 22

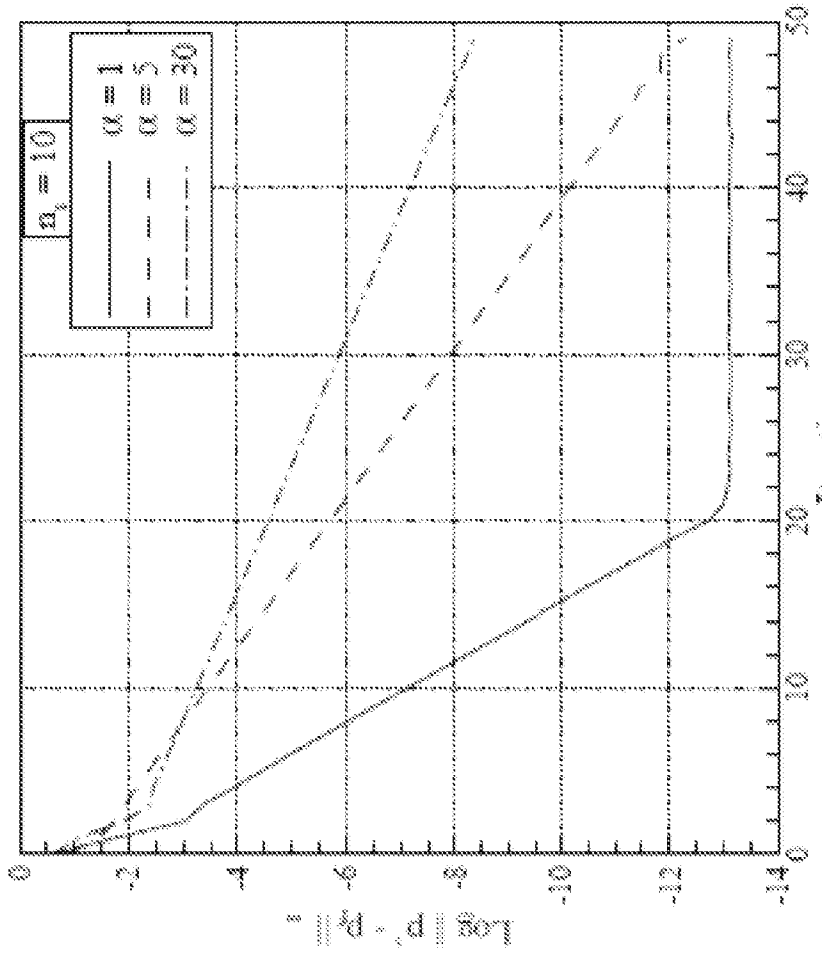


FIG. 23B

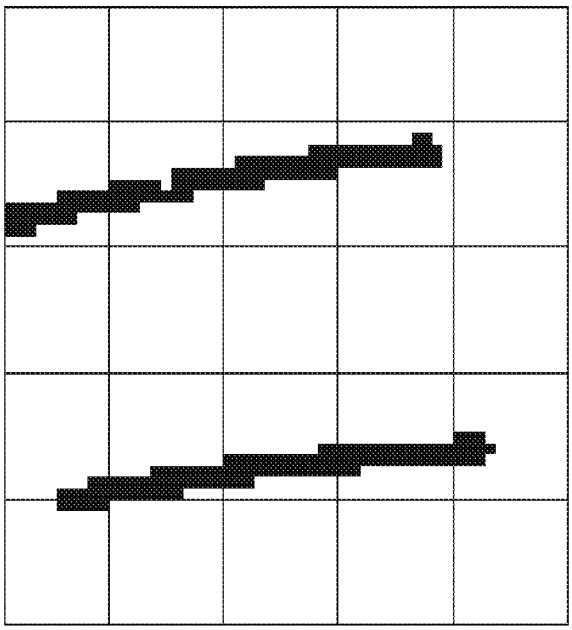


FIG. 23A

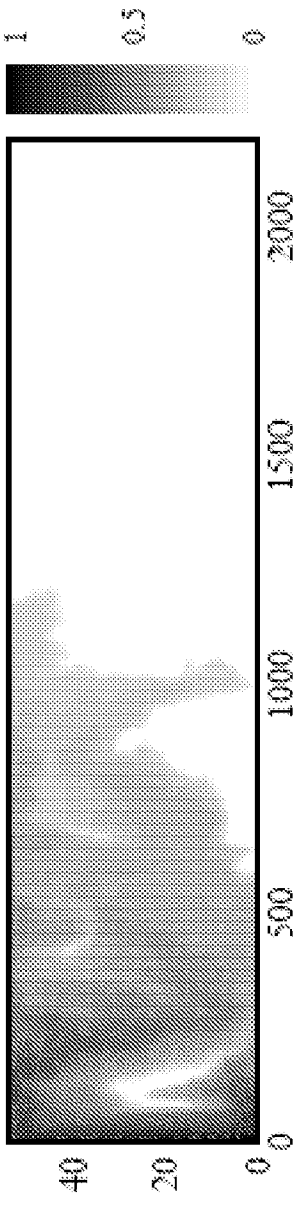


FIG. 24A

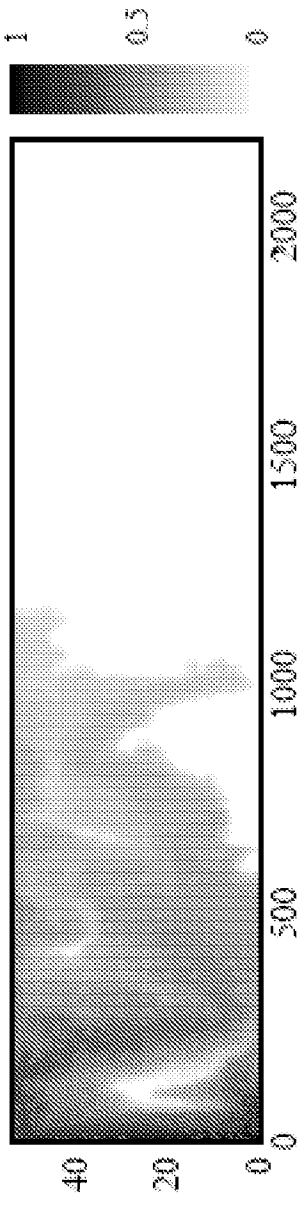


FIG. 24B

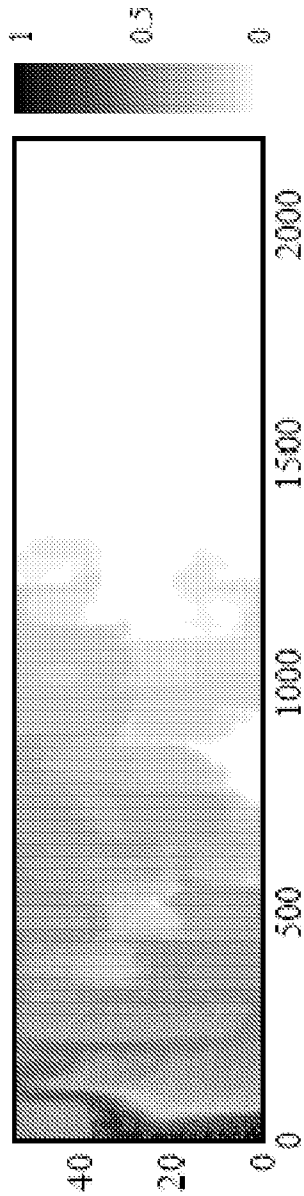


FIG. 24C

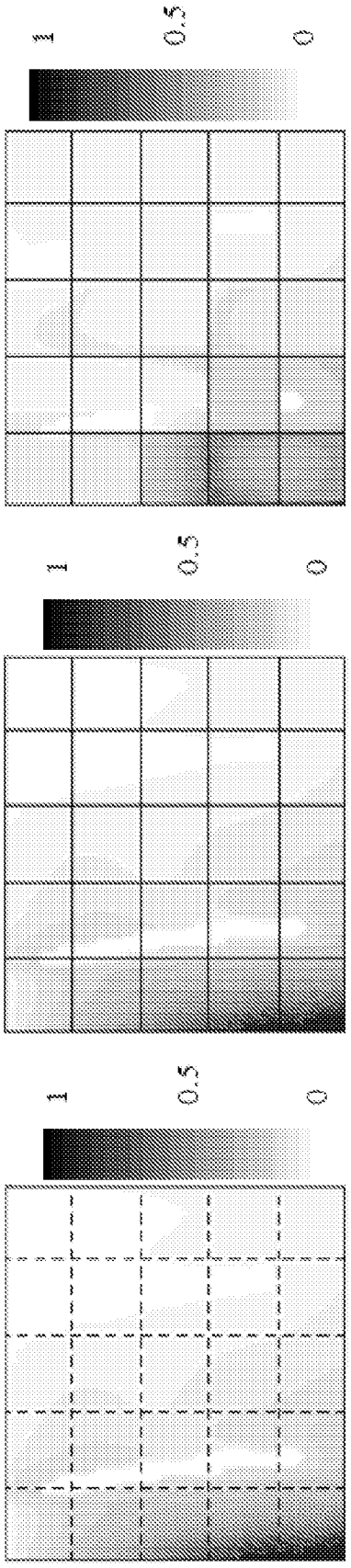


FIG. 25C

FIG. 25B

FIG. 25A

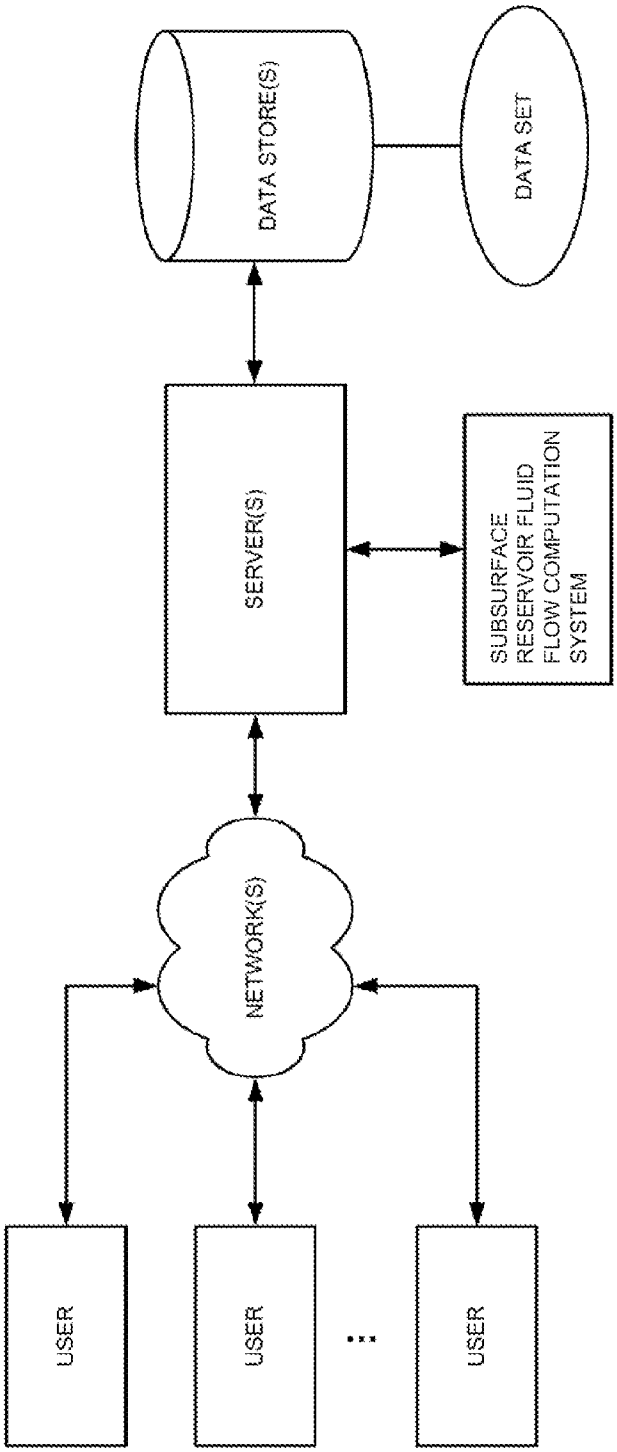

DATA STORE(S)
DATA SET
SERVER(S)
SUBSURFACE RESERVOIR FLUID FLOW COMPUTATION SYSTEM
NETWORK(S)
USER
USER
...
USER

FIG. 26