Mathematics, parallel computing and reservoir simulation

Petter E. Bjørstad *

Abstract

This paper will highlight, by way of examples, a few seemingly very different mathematical problems and show how they have direct relevance to the construction of efficient computational procedures for the simulation of oil reservoirs on parallel computers.

1 Partitioning of graphs

Consider a graph $G(V, E)$, having a set of vertices connected by edges. We associate work with each vertex and communication (or dependencies) with each edge. Our problem is to partition the graph into two parts such that the two sets are similar in size. We further want the dependencies (i.e., the number of edges) between the two sets to be as small as possible.

Mathematically this can be expressed by having a variable x_i at vertex V_i be one or minus one depending on which of the two sets V_i belongs to. A little reflection shows that our problem can be formulated

$$\min \sum_{E_{ij}} (x_i - x_j)^2 \quad \text{subject to} \quad \sum_i x_i = 0, \; x_i = \pm 1.$$

Define the degree d_i of vertex V_i as the number of edges connected to that vertex. We can formulate the minimization problem as an equivalent matrix problem by defining the **Laplacian** matrix of the graph G by

$$L_{ij} = \begin{cases} d_i & \text{if} \quad i = j \\ -1 & \text{if} \quad i, j \in E_{ij} \\ 0 & \text{otherwise} \end{cases}$$

*Institutt for Informatikk, Høyteknologisenteret, University of Bergen, N-5020 Bergen, Norway. (http://www.ii.uib.no/~petter, Email: petter@ii.uib.no)

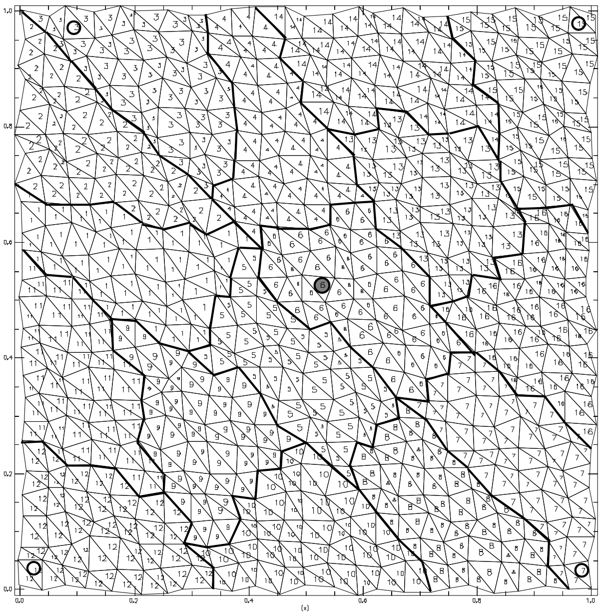

Figure 1: A simple oil reservoar model with four injection wells and one production well. The discrete model has been partitioned into 16 subdomains using the method in Section 1.

It is straightforward to show that our problem is equivalent with

$$\min \ x^T L x \quad \text{subject to} \quad x^T e = 0 \ \text{ and } \ x_i = \pm 1,$$

where e is a vector of all ones. This problem is NP-hard, we therefore consider the problem without the discrete constraint on x, that is, we require instead that $x^T x = n$, where n is the size of our problem. Clearly, this minimum must be a lower bound for our original problem since we minimize over a larger space. In order to solve the continuous problem we note that the matrix L is symmetric, semi-definite and that e is an eigenvector corresponding to $\lambda_1 = 0$. Since $x^T e = 0$, we can expand the solution $x = \sum_{i=2}^{n} a_i v_i$ in the set of orthogonal eigenvectors of L, and find by direct computation that

$$x^T L x = \sum_{i=2}^{n} a_i^2 \lambda_i \geq n\lambda_2,$$

since $x^T x = \sum_{i=2}^{n} a_i^2 = n$. Moreover, $x = \sqrt{n} v_2$ achieves this lower bound and therefore solves our problem.

The problem can therefore be solved by computing the eigenvector corresponding to the smallest positive eigenvalue of L. Very efficient algorithms for this problem are known.

It remains to find a good approximation to our original discrete problem. The closest discrete solution to our continuous solution can be easily constructed by computing the median value of the components x_i and then assign -1 to all components having a lower value and the value $+1$ to the complement set.

Many refinements of the simple strategy outlined here are possible. The interested reader is referred to the recent journal article by Hendrickson and Leland [8] and the references to found there for a more extensive study of these techniques. This exposition is a greatly simplified version of their more comprehensive and general theory. The software package Chaco [7] was used to produce the partioning of the grid in Figure 1.

2 Approximation and stability properties of the weighted L^2 projection

Our next problem is concerned with L^2 projections and their mathematical properties. Consider the model problem

Find $u^* \in V(\Omega)$ such that

$$a(u^*, v) = f(v) \quad \forall v \in V, \tag{1}$$

with

$$a(u, v) = \sum_{i=1}^{N} \int_{\Omega_i} \rho_i \nabla u \cdot \nabla v \; dx$$

and

$$f(v) = \int_{\Omega} f v \; dx,$$

in an appropriate Sobolev space $V(\Omega)$. Note that our problem domain Ω is the union of subdomains Ω_i and that the parameter ρ_i can change across subdomains. Here ρ_i are positive constants and $\Omega = \cup_{i=1}^{N} \Omega_i$.

In practice we will compute an approximate solution to the above problem. Typically, we define a subspace $V_h \subset V$ and search for a solution $u \in V_h$ such that

$$a(u, v) = f(v) \quad \forall v \in V_h.$$

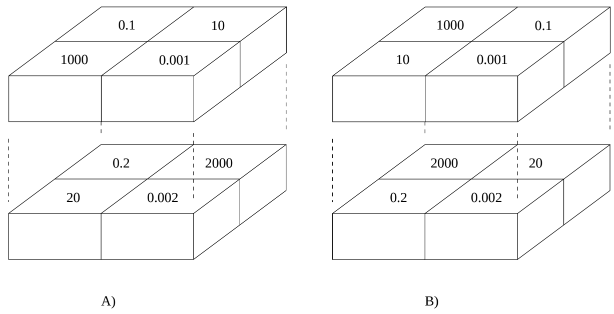

Figure 2: Non-quasimonotone (A) and quasimonotone (B) distribution of the coefficients ρ_i.

That is, we find $u \in V_h$ as an ortogonal projection (in the $a(\cdot,\cdot)$ inner product) of the solution $u^* \in V$ onto the subspace V_h. Similarly, corresponding to the subdomains Ω_i we define a coarse subspace V_H, where H is the diameter of the largest subdomain. In the regular case one can view $V_H \subset V_h$ as the space of functions that results from a coarse discretization of (1) using the subdomains as discretization elements of Ω.

Given $u \in V_h$, define the function $u_0 = Q_H^\rho u$ such that

$$(u_0, v) = (u, v) \quad \forall v \in V_H,$$

with

$$(u, v) = \sum_{i=1}^{N} \int_{\Omega_i} \rho_i u v \, dx,$$

a weighted L^2 inner product. In light of the definition above u_0 is the weighted L^2 projection of u on V_H.

Consider first the case when $\rho_i = 1$ corresponding to the normal L^2 inner product. The L^2 projection has a reasonable approximation property as can be seen from the following estimate (p. 165 in [9]):

$$||u - u_0||_{L^2(\Omega)} \leq CH|u|_{H^1(\Omega)},$$

where the seminorm $|u|_{H^1(\Omega)} = a(u,u)^{1/2}$. From this estimate one can also prove H^1 stability of the projection, that is, $|u_0|_{H^1(\Omega)} \leq C|u|_{H^1(\Omega)}$. In other words, the energy of the projection $u_0 = Q_H u$ is bounded by the energy of the original function u.

When the coefficients ρ_i are allowed to have jumps from one subdomain to the next, the situation becomes more involved [4], [6], [10]. In fact, it has been shown that in the general case (see Figure 2A), the approximation property can only be estimated by:

$$\sum_{i=1}^{N} \rho_i ||u - u_0||^2_{L^2(\Omega_i)} \leq CH^2 \frac{H}{h} \sum_{i=1}^{N} \rho_i |u|^2_{H^1(\Omega_i)}, \tag{2}$$

while if the coefficients have a so-called quasimonotone distribution (see Figure 2B), we recover the estimate for the case $\rho_i = 1$,

$$\sum_{i=1}^{N} \rho_i ||u - u_0||^2_{L^2(\Omega_i)} \leq CH^2 \sum_{i=1}^{N} \rho_i |u|^2_{H^1(\Omega_i)}. \tag{3}$$

A quasimonotone distribution is verified by being able to traverse a path from any subdomain (across common faces) through neighboring subdomains (sharing a common vertex as in Figure 2) such that the value of the coefficient is monotone increasing to the subdomain with the largest value. We observe that the approximation property may deteriorate like $1/h$ in estimate (2), similarly we lose H^1 stability, the energy of the projection may exceed the energy of the original function u.

3 The oil reservoir problem

We now give a brief description of our application, the simulation of flow in porous media. The problem carries substantial economic interest in the petroleum industry. It is widely used for planning purposes, for reservoir management and for prediction of the reservoir performance [2]. Another field of application is the study of ground water flow, in particular, the study of ground contamination by way of pollution simulation models [1].

A petroleum reservoir consists of hydrocarbons and other chemicals trapped in tiny pores in the rock. If the rock permits and if the fluid is sufficiently forced, it will flow from its current position in the rock to some other points in the reservoir. By injection of additional fluids and the release of pressure through the production of fluids at wells, the petroleum engineer can adjust the flow rate and modify the mixture of chemicals produced. In the associated reservoir flow problem, we are given the reservoir conditions (pressure and saturation) and the well flow conditions. The main problem is to model the fluid flow, especially to predict the fluid flow into production wells. The fluid flow problem can be expressed in terms of approximate mathematical equations. The physical laws that govern the flow can be derived from volume balance, phase equilibrium and conservation of mass, plus

Darcy's Law stating that the fluid flow is proportional to pressure gradients and gravitational potential differences [1]. The constants of proportionality depend on permeability, viscosity and the density of the phases involved. The equations constituting a mathematical model for a reservoir are almost always too complex to be solved by analytical methods, even after many idealizations.

A simplified *prototype* black oil simulator [5] with incompressible flow and neglecting gravity can be reduced to an equation for the pressure

$$-\nabla \cdot C(s,x) \cdot \nabla p = q_1(x,t), \tag{4}$$

and a saturation equation

$$\phi \frac{\partial s}{\partial t} + v(x) \cdot \nabla f(s) - \epsilon \nabla \cdot (D(s,x) \cdot \nabla s) = q_2(x,t). \tag{5}$$

There are many different approaches for solving the equations. One can treat the complete nonlinear system using Newton's method [2] or decouple the system and iterate between the pressure equation and the saturation equation. The pressure equation is solved with the saturation fixed, and then the saturation equation is solved with the velocity (pressure) fixed. This splitting is not supported by a rigorous mathematical theory; rather it is heuristically motivated by the different behavior of the pressure equation and the saturation equation [1].

4 Domain decomposition

Next, we illustrate the importance and impact of the mathematical examples from the first two sections applied to the pressure equation (4). We note that this equation has a weak (variational) formulation similar to (1). We assume that the function $C(s,x)$ is piecewise constant ($= \rho_i$) in subdomain Ω_i. This function reflects material properties, in particular, the permeability of the rock. The permeability in reservoirs may change by several orders of magnitude (across rock interfaces or layers) in a rather discontinuous manner. Our formulation in (1) is therefore often of interest.

The discrete version of (4) will produce a very large linear system of equations

$$Ax = b.$$

The matrix A typically inherits properties from the continuous problem and is symmetric and positive definite. Effective iterative algorithms for the solution of this equation are most often based on the idea of preconditioning, that is, the approximate construction of an inverse matrix B. One can then apply a basic iterative procedure of the form

$$x^{k+1} = x^k + B(b - Ax^k),$$

and accelerate this with a suitable Krylov subspace method [9].

The success of this approach depends on how well one can construct the matrix B. Domain decomposition methods try to construct B from exact or approximate inverses that are localized, that is, they approximate the inverse of A restricted to a subdomain.

Assume that we partition the original discrete problem using the techniques described in the first section. In this way we can define a set of subdomains Ω_i, with a balanced size and with a small number of interconnecting nodes. (The procedure in Section 1 defined two sets, but we can obviously apply this algorithm recursively, more refined techniques can further be used to define just the number of subdomains that we require.) Define the (rectangular) restriction matrix R_i applied to a vector x defined on all nodes, to return just the coefficients that belong to subdomain Ω_i. The local matrix associated with Ω_i is $A_i = R_i A R_i^T$, and a corresponding local inverse can be expressed as

$$B_i = R_i^T (R_i A R_i^T)^{-1} R_i.$$

In addition, these methods almost always need a representation that can capture the smooth or average behavior of the solution. This should be a matrix of much lower dimension than the original A, its role can equivalently be seen as a global coupling of all the localized approximations. If the subdomain decomposition forms a regular grid then this matrix is often just the stiffness matrix derived from a discretization where the subdomains are considered to be the basic discretization elements. We denote the inverse of such a low dimensional, coarse space approximation B_C.

A simple additive preconditioner can now be constructed by taking

$$B = B_C + \sum_i B_i.$$

In order for this to work well, we need to make an important modification of the approach outlined above. We must introduce a better coupling between the local spaces and the coarse space. This is often done by making

Method	Additive Average			Additive Diagonal		
$\rho = 1$	39	(28.3)	204	28	(14.0)	574
Quasi-monotone	44	(31.0)	221	34	(15.1)	592
Not quasi-monotone	39	(20.9)	201	136	(363)	1028

Table 1: Comparison of two iterative methods. The number of iterations required to reduce the residual by six orders of magnitude is followed by the condition number of the preconditioned system in parenthesis. The last number given is the elapsed time in seconds when using a cluster of 8 Ultra Sparc processors. All subdomain and coarse problems are solved approximately by using only two symmetric Gauss-Seidel iterations.

the subdomains overlap each other. This modification produces the so-called Additive Schwarz preconditioners.

An alternative to overlap is to change or complement the coarse space matrix B_C. We briefly discuss two such alternatives where we avoid overlap. (Thus, we just keep the subdomains given by our graph partitioning algorithm) [3].

In the first method, called an Additive Diagonal Scaling method, we add yet another special matrix to our sum above. This matrix, which is diagonal, arises from the restriction of the space V_h to the union of all interior subdomain interfaces and the use of an approximate bilinear form where we just sum products $u(x)v(x)$ for nodal values x on the interface. That is, we solve a diagonal problem for unknowns corresponding to all interior interface nodes.

In the second method, called an Additive Average method, we define a completely new B_C that includes all the interface variables, but in a way that leaves us with a system that is relatively easy to solve. This matrix corresponds to a new coarse space defined as the range of an interpolation-like operator that extends into the interior of each subdomain by just using the average value of the subdomain boundary nodal values (see [3] for details).

In Table 1 we compare the two methods for different distributions of the coefficient ρ_i. The computational example has 512 regular subdomains in three space dimensions, with a total of approximately a quarter million of unknowns. The time per iteration was about 4.2 second for all cases, however, the construction of the coarse part, B_C, of the preconditioner takes considerable time for the Additive Diagonal method. We immediately notice the poor behavior of the Additive Diagonal method in the non quasi-monotone case. It is, at this point, perhaps not surprising that this difference is directly related to the mathemathical properties of the L^2 projection discussed in Section 2. The analysis of the Additive Diagonal method depends on the approximation and H^1 stability properties of this projection. However, by constructing a completely different coarse space as we do in the Additive Average method, we are able to avoid this difficulty and indeed work with an interpolation operator which has the desired approximation and stability properties also in the general case where the ρ_i may have a non quasi-monotone distribution.

5 Concluding notes and remarks

We have seen how a discrete optimization problem can be approximated by a continuous problem which in turn reduces to the solution of a very special eigenvalue problem. Furthermore, this problem has important applications to the design of parallel computer algorithms, since it can be used to partition a problem among many processors in such a way that each processor receives a fair share of the work, with relatively little communication to other processors.

Also, we have outlined how quite abstract and subtle estimates of approximation and stability properties of the weighted L^2 projection shows up as the difference between 33 and 134 iterations in a numerical algorithm applied to an oil reservoir simulation problem. Furthermore, how such insights may help in the construction of numerical algorithms that are more robust with respect to large, discontinuous variations in, for example, the permeability of rock.

Mathematical techniques and knowledge spanning across a very broad area are needed in order to build a practical numerical algorithm that effectively can be used as a part in an even more complex computer simulation package for todays challenging engineering problems.

Acknowledgment

The author thanks Eero Vainikko for providing the computational results given in Table 1.

References

[1] M. B. Allen, G. A. Behie, and J. A. Trangenstein, *Multiphase Flow in Porous Media*, Springer-Verlag, 1988. Lecture Notes in Engineering.

[2] K. Aziz and A. Settari, *Petroleum reservoir simulation*, Elsevier Applied Science Publishers, 1979.

[3] P. Bjørstad, M. Dryja, and E. Vainikko, *Additive schwarz methods with no subdomain overlap and new coarse spaces*, in Domain Decomposition Methods in Sciences and Engineering, R. Glowinski, J. Périaux, Z. Shi, and O. B. Widlund, eds., John Wiley & Sons, 1996, pp. xx–yy. Proceedings from the Eight International Conference on Domain Decomposition Metods, May 1995, Beijing.

[4] J. H. Bramble and J. Xu, *Some estimates for a weighted L^2 projection*, Math. Comp., 56 (1991), pp. 463–476.

[5] G. Chavent, G. Cohen, and J. Jaffra, *Discontinuous upwinding and mixed finite elements for two-phase flows in reservoir simulation*, Comp. Meth. in Appl. Mech and Eng., 47 (1984), pp. 93–118.

[6] M. Dryja, M. Sarkis, and O. B. Widlund, *Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions*, Tech. Report 662, Department of Computer Science, Courant Institute, March 1994.

[7] B. Hendrickson and R. Leland, *The Chaco user's guide, version 2.0*, Tech. Report SAND 94-2692, Sandia National Laboratories, July 1995.

[8] ——, *An improved spectral graph partitioning algorithm for mapping parallel computations*, SIAM J. Sci. Comput., 16 (1995).

[9] B. F. Smith, P. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.

[10] J. Xu, *Counter examples concerning a weighted L^2 projection*, Math. Comp., 57 (1991), pp. 563–568.