SANDIA REPORT

SAND2012-4460
Unlimited Release
Printed May 2012

Preconditioner and Convergence Study for the Quantum Computer Aided Design (QCAD) Nonlinear Poisson Problem Posed on the Ottawa Flat 270 Design Geometry

Irina Kalashnikova

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2012-4460
Unlimited Release
Printed May 2012

Preconditioner and Convergence Study for the Quantum Computer Aided Design (QCAD) Nonlinear Poisson Problem Posed on the Ottawa Flat 270 Design Geometry

Irina Kalashnikova
Numerical Analysis and Applications Department
Sandia National Laboratories
P.O. Box 5800, MS 1320
Albuquerque, NM 87185-9999
ikalash@sandia.gov

Abstract

A numerical study aimed to evaluate different preconditioners within the Trilinos `Ifpack` and `ML` packages for the Quantum Computer Aided Design (QCAD) non-linear Poisson problem implemented within the Albany code base and posed on the Ottawa Flat 270 design geometry is performed. This study led to some new development of Albany that allows the user to select an `ML` preconditioner with `Zoltan` repartitioning based on nodal coordinates, which is summarized. Convergence of the numerical solutions computed within the QCAD computational suite with successive mesh refinement is examined in two metrics, the mean value of the solution (an L^1 norm) and the field integral of the solution (L^2 norm).

Contents

Figures

Tables

1 Introduction

The Quantum Computer Aided Design (QCAD) project, funded by the LDRD program (project #151297), aims to develop computational tools that enable the design and analysis of few-electron, low-temperature quantum devices at specific regimes, and facilitate the discovery of quantum dot structures [2]. The project uses as leverage the high-performance parallel computing resources available at Sandia, as well as a number of existing Sandia software tools, including the Trilinos solver library, the Dakota optimization toolbox, the Cubit mesher, and the SIERRA toolkit. QCAD is implemented in the Albany code base, under the Agile Components strategy of Trilinos. Within this framework, a suite of computational tools that focuses around the efficient numerical solution of the non-linear Poisson equation and coupled Poisson-Schrödinger equations has been developed.

In recent months, the QCAD computational suite has been adopted by analysts to help with the design of actual quantum devices to be constructed in a lab. Since these analysts require timely and accurate predictions to aid their designs, there is much motivation to improve the code's performance without sacrificing any robustness. There is also motivation to validate the solutions computed within the code. Toward this effect, the present document summarizes the results of a numerical study aimed to address the following questions:

1. Which choice of preconditioner within the Trilinos `Ifpack` (ILU) [5] and `ML` (algebraic multi-grid) [3] packages minimizes the total solve time of the linear systems ($\mathbf{Ax} = \mathbf{b}$) that arise from the discretization of the governing equations? (Section 2)

2. Do the solutions computed within the QCAD computational suite converge with successive mesh ($h-$) refinement? (Section 3)

In answering these questions, opportunities to identify ways to improve the Albany code base through new development were sought. The first part of the numerical study led to the development of a new capability within Albany that allows the user to select an `ML` preconditioner with `Zoltan` repartitioning based on nodal coordinates. This choice of preconditioner can greatly accelerate convergence for large problems run on a large number of processors, and could benefit numerous applications implemented within Albany. The second part of the numerical study was of particular interest due to the nature of the Cubit-generated tetrahedral meshes used to discretize the quantum devices that are of interest. The irregular shape of these devices (Figure 1) makes them difficult to mesh. This reality makes the generation of "bad" meshes a genuine possibility. A numerical convergence study with respect to mesh refinement would validate in some sense the meshing algorithm employed as well as put confidence in the solutions obtained from the QCAD computational suite.

In this document, attention is restricted to the "Ottawa Flat 270" device design (Figure 1), a modification of the original Ottawa device design [4]. The device is discretized using four-node tetrahedral finite elements, generated using the Cubit meshing tool. The governing partial differential equation (PDE) is a non-linear Poisson equation, which describes the large population of atoms/molecules of which a device is composed.

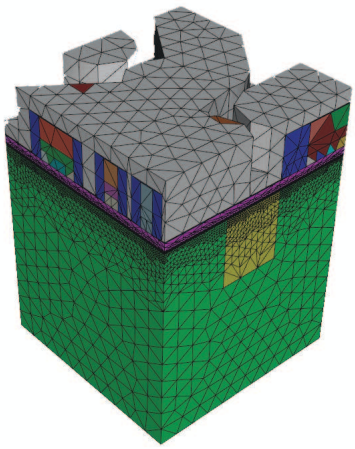
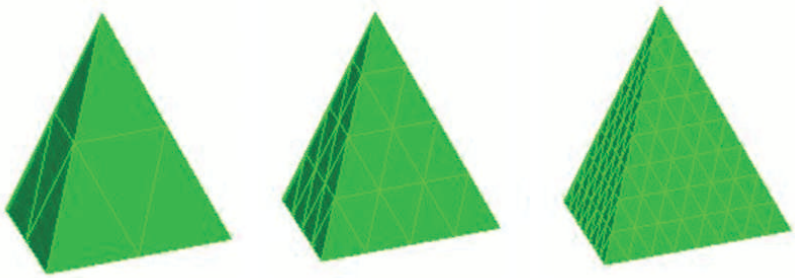


Figure 1. Sample quantum device model

The remainder of this document is organized as follows. Section 2 describes the `Ifpack` and `ML` preconditioners considered in the preconditioner performance study, as well as the new development within Albany that was motivated by this study. Total linear solve and total preconditioner creation times[1] for each preconditioner considered are reported for problems of three sizes: a "small" problem of ≈ 1 million tetrahedral elements, a "medium" problem of ≈ 8.5 million tetrahedral elements, and a "large" problem of ≈ 69 million tetrahedral elements. The two finer meshes were generated through a successive mesh refinement of the original ≈ 1 million four-node tetrahedral "coarse" mesh, that is, by splitting all elements in the original "coarse" mesh evenly in all directions. This successive mesh refinement can be achieved in Cubit using the command:

```
refine volume all numsplit N
```



(a) Original "coarse" mesh **(b)** `numsplit 1` **(c)** `numsplit 2`

Figure 2. Mesh refinement

where N is the level of successive mesh refinement (e.g., if $N = 1$ each element will be refined once in each direction, if $N = 2$ each element will be refined twice in each direction, etc; Figure 2). All runs were performed on the 160-TFlop Red Sky cluster at Sandia. The "small", "medium"

[1] Included in the total linear solve time.

and “large” problems were run in parallel on 16 processors (2 nodes), 128 processors (16 nodes) and 1024 processors (128 nodes) respectively. These processor and node counts were selected so that for each problem size, each processor had the same number of elements, and all eight processors of each node were occupied. Scalability with respect to problem size and processor count is examined.

Section 3 presents the results of a numerical convergence study of the QCAD solution with respect to successive mesh refinement, performed in the manner described above. The convergence of two quantities of interest, the mean value of the solution and the field integral (the integral of the solution over the domain), was considered specifically. The observed rates of convergence of these quantities are compared to the expected theoretical convergence rates. Conclusions are offered in Section 4.

2 Preconditioner Performance Study

This section summarizes the results of a study aimed at evaluating the relative performance of different preconditioners available through the Trilinos `Ifpack` and `ML` packages for a QCAD nonlinear Poisson problem posed on the Ottawa Flat 270 device geometry. Twelve basic preconditioner types are considered: nine `Ifpack` preconditioners and three `ML` preconditioners (Table 1). The `Ifpack` preconditioners are effectively ILU preconditioners, and differ in the `overlap` and `level-of-fill` options. The `ML` preconditioners are algebraid multi-grid preconditioners based on three default preconditioner types available in the `ML` package: `SA` (classical Smoothed Aggregation), `DD` (classical smoothed aggregation based on two-level Domain Decomposition), and `DD-ML` (three-level algebraic Domain Decomposition). For a detailed discussion of these `Ifpack` and `ML` options, the reader is referred to the `Ifpack` and `ML` user guides, [5] and [3] respectively.

Table 1. Summary of preconditioners evaluated

Preconditioner #	Type	Parameters
1	ifpack	overlap = 0, level-of-fill = 0
2		overlap = 1, level-of-fill = 0
3		overlap = 2, level-of-fill = 0
4		overlap = 0, level-of-fill = 1
5		overlap = 1, level-of-fill = 1
6		overlap = 2, level-of-fill = 1
7		overlap = 0, level-of-fill = 2
8		overlap = 1, level-of-fill = 2
9		overlap = 2, level-of-fill = 2
10	ML	default type = SA
11		default type = DD
12		default type = DD-ML

By perusing the `ML` users' guide [3], the reader may observe that the `ML` preconditioner package has a number of options and parameters that may be specified by the user, and/or over-written from the default settings. In an effort to optimize the performance of the `ML` preconditioners, it is worthwhile to explore several of these options. To this effect, three variants of the `ML` preconditioners introduced in Table 1, referred to as A, B and C, are considered. The parameter lists for these preconditioner options are summarized in Table 2 for the specific case of an `SA` default preconditioner. The C variant preconditioner employs the matrix repartitioning option available through the Trilinos `Zoltan` package. Essentially, repartitioning uses information about the mesh coordinates to perform dynamic load-balancing of coarse-level matrices in the multigrid preconditioner. With repartitioning, message passing latency on the coarse level can be improved, and the well-known problem of the coarsening rate dropping as the number of unknowns per processor becomes small can be avoided. Providing the user with the option to select an `ML` preconditioner with `Zoltan` repartitioning required some non-trivial new development within Albany. Functions that identified and communicated the (x,y,z) coordinates of a problem's underlying mesh were added to existing Albany classes.

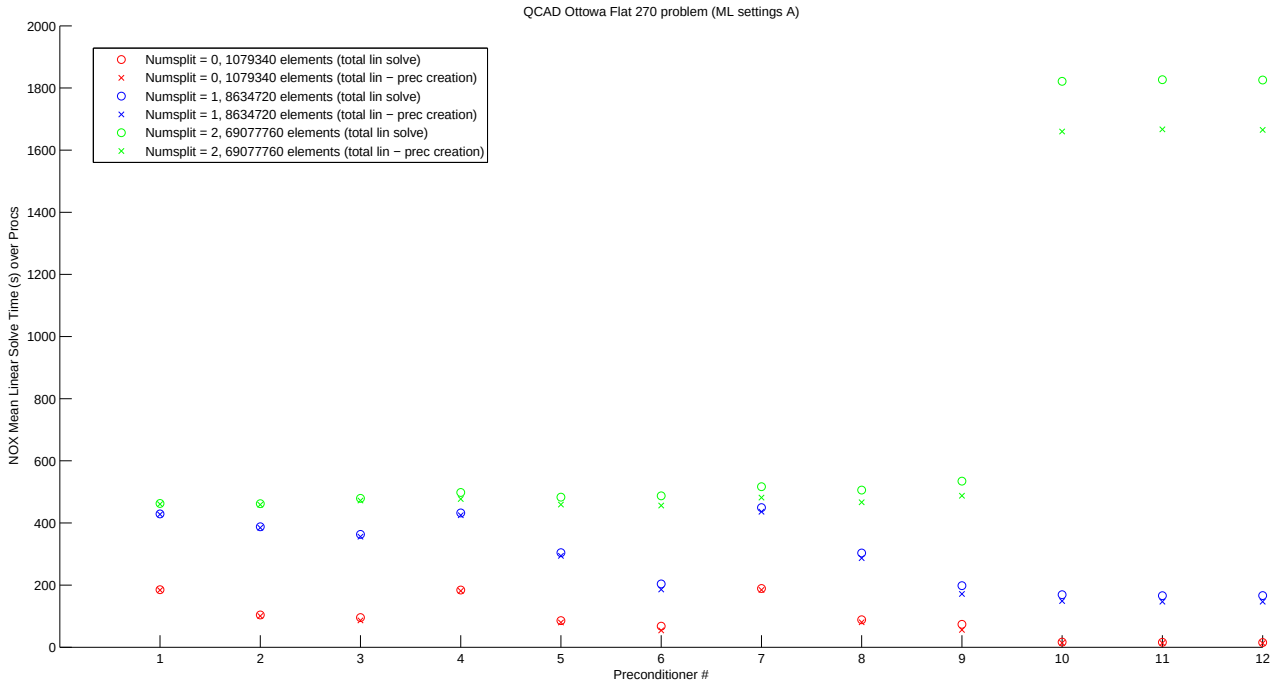


Figure 3. `Ifpack` vs. `ML` preconditioners with `ML` settings A

Figures 3–5 depict the `Belos` total linear solve times and total preconditioner creation times for the nine `Ifpack` preconditioners and the three `ML` preconditioners summarized in Table 1. The `ML` preconditioners in Figure 3 have the `ML` settings A; the `ML` preconditioners in Figure 4 have the `ML` settings B; the `ML` preconditioners in Figure 5 have the `ML` settings C. With settings that are effectively the default `ML` settings (settings A), the `ML` preconditioners are outperformed by the `Ifpack` preconditioners by a large margin on the finest mesh considered (Table 3). The performance of the `ML` preconditioners improves when the `aggregation: type` is changed to `Uncoupled-MIS` (settings B); however the `ML` preconditioners still do not outperform the `Ifpack` preconditioners on the finest mesh (Figure 4). Inspection of the verbose output from the `ML` package suggested that the situation may be improved by introducing `Zoltan` repartitioning based on nodal coordinate, and the Albany code base was modified to allow this option, as discussed above. The reader may observe an extraordinary speedup in the total linear solve and preconditioner creation times for the `ML` preconditioners with repartitioning (settings C) (Figure 5). With settings C, the `ML` preconditioners achieve a factor speedup of more than two relative to the `Ifpack` preconditioners for *all* mesh resolutions considered. More specifically, for a problem discretized by ≈ 8.6 million tetrahedral elements, run on 128 processors on the Red Sky cluster:

- The linear solves were 2.4–5.45 times faster with an `ML` preconditioner plus `Zoltan` repartitioning (settings C) compared to an `Ifpack` preconditioner.
- The linear solves with an `ML` preconditioner plus `Zoltan` repartitioning (settings C) were ≈ 2 times faster than with a "black box" `ML` preconditioner (settings A).

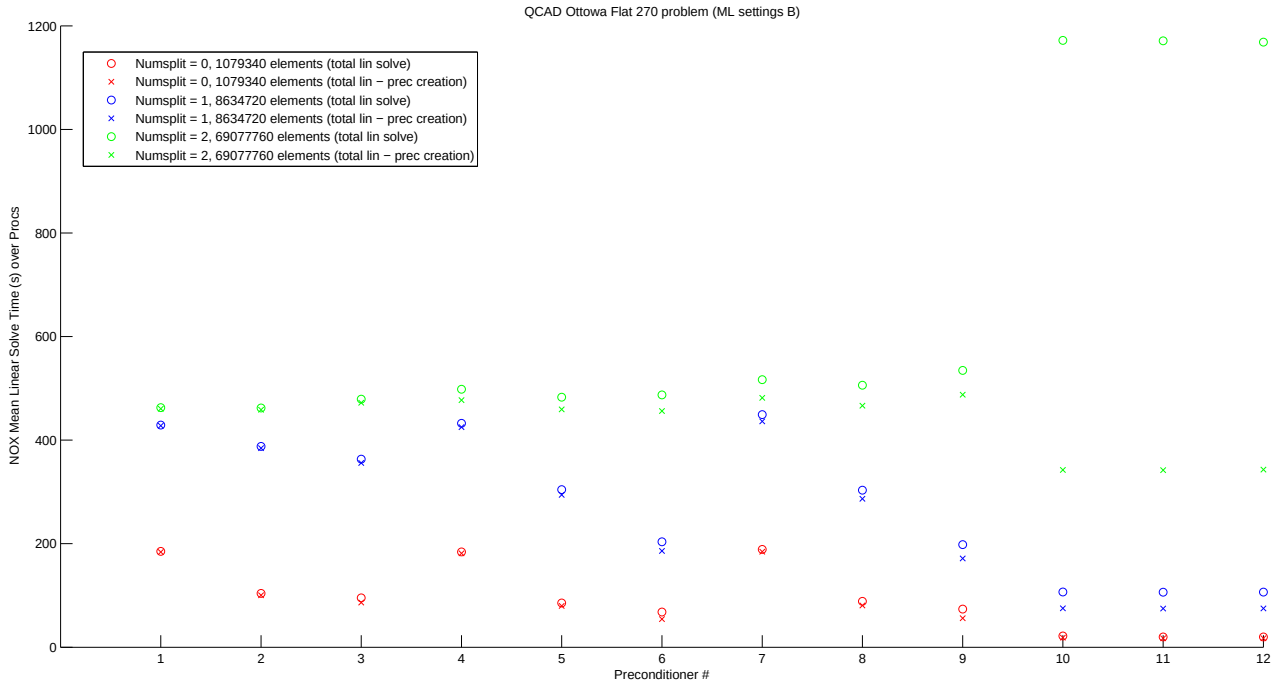


Figure 4. `Ifpack` vs. `ML` preconditioners with `ML` settings B

For a problem discretized by ≈ 64 million tetrahedral elements, run on 1024 processors on the Red Sky cluster:

- The linear solves were 2.4–2.79 times faster with an `ML` preconditioner plus `Zoltan` repartitioning (settings C) compared to an `Ifpack` preconditioner.
- The linear solves with an `ML` preconditioner plus `Zoltan` repartitioning (settings C) were 9.5 times faster than with a "black box" `ML` preconditioner (settings A)

The `ML` preconditioner option with `Zoltan` repartitioning (settings C) is therefore recommended for all problem sizes.

Figure 6 shows the preconditioner number (Table 1) versus

$$\frac{\text{total linear solve time}}{\text{\# elements per processor}}, \tag{1}$$

for the three problem sizes considered. If the problem scaled perfectly with the number of processors, the value (1) would be the same for all problem sizes and processor counts. Figure 6 indicates that scalability for this problem could be improved. The reason for the suboptimal scalability demonstrated in this figure is at the present time unknown, and may be investigated in future work.

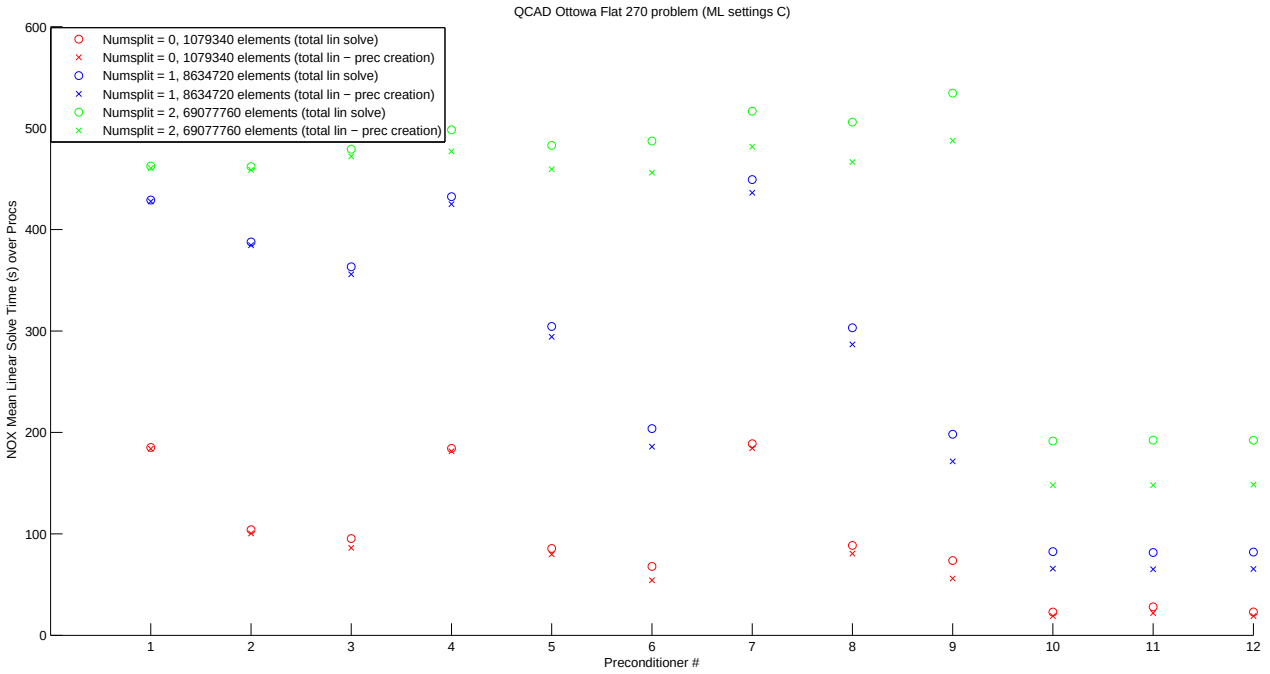


Figure 5. `Ifpack` vs. `ML` preconditioners with `ML` settings C

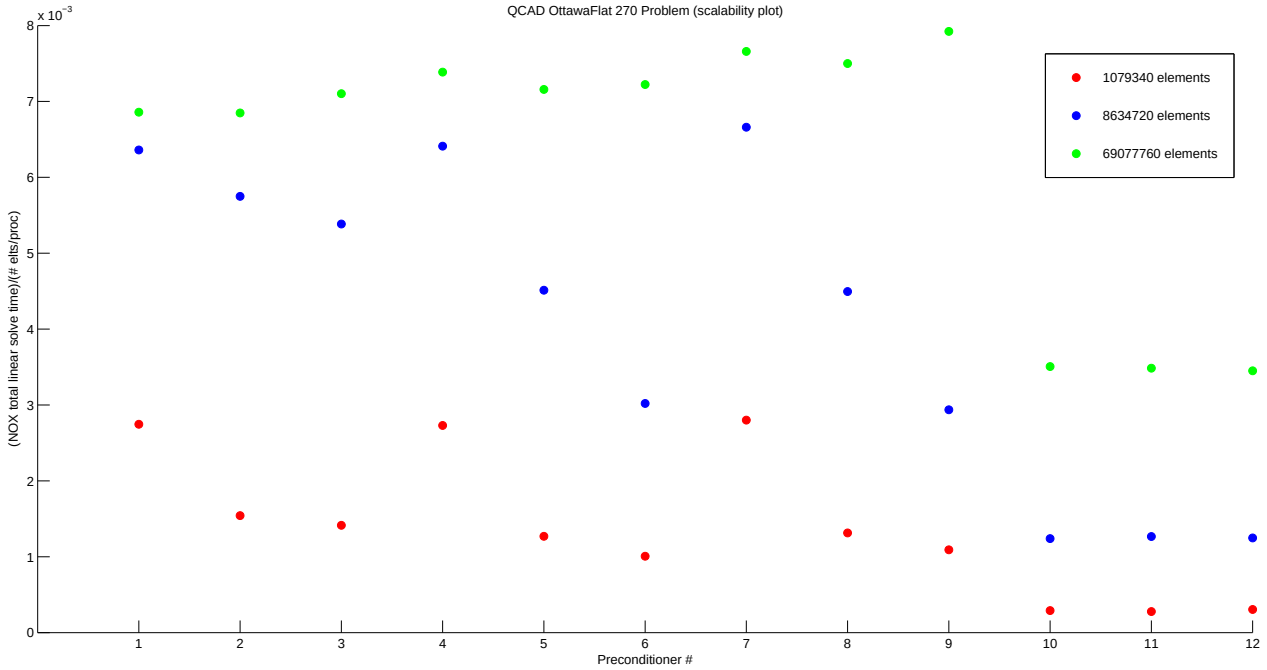


Figure 6. Scalability plot

Table 2. Summary of ML settings evaluated (for example of `default values: SA`)

ML settings A

```
<ParameterList name="ML">
<Parameter name="Base Method Defaults" type="string" value="none"/>
   <ParameterList name="ML Settings">
      <Parameter name="default values" type="string" value="SA"/>
     <Parameter name="smoother:  type" type="string" value="Chebyshev"/>
     <Parameter name="smoother:  pre or post" type="string" value="both"/>
     <Parameter name="coarse:  type" type="string" value="Amesos-KLU"/>
   </ParameterList]>
</ParameterList>
```

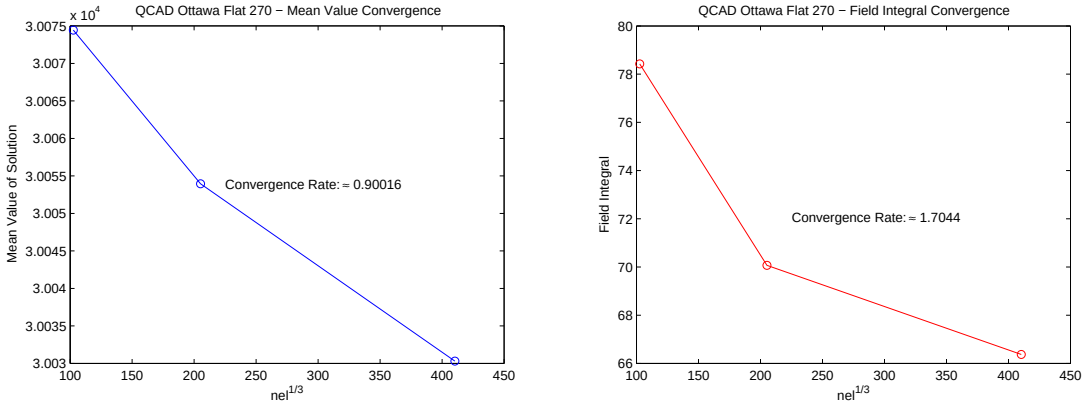
ML settings B

```
<ParameterList name="ML">
<Parameter name="Base Method Defaults" type="string" value="none"/>
  <ParameterList name="ML Settings">
     <Parameter name="default values" type="string" value="SA"/>
     <Parameter name="smoother:  type" type="string" value="Chebyshev"/>
     <Parameter name="smoother:  pre or post" type="string" value="both"/>
     <Parameter name="coarse:  type" type="string" value="Amesos-KLU"/>
     <Parameter name="coarse:  max size" type="int" value="512"/>
     <Parameter name="aggregation:  type" type="string" value="Uncoupled-MIS"/>
  </ParameterList>
</ParameterList>
```

ML settings C

```
<ParameterList name="ML">
<Parameter name="Base Method Defaults" type="string" value="none"/>
  <ParameterList name="ML Settings">
     <Parameter name="default values" type="string" value="SA"/>
     <Parameter name="smoother:  type" type="string" value="Chebyshev"/>
     <Parameter name="smoother:  pre or post" type="string" value="both"/>
     <Parameter name="coarse:  type" type="string" value="Amesos-KLU"/>
     <Parameter name="coarse:  max size" type="int" value="512"/>
     <Parameter name="repartition:  enable" type="int" value="1"/>
     <Parameter name="repartition:  partitioner" type="string" value="Zoltan"/>
     <Parameter name="repartition:  Zoltan dimensions" type="int" value="3"/>
     <Parameter name="repartition:  max min ratio" type="double" value="1.3"/>
     <Parameter name="repartition:  min per proc" type="int" value="1000"/>
   </ParameterList>
</ParameterList>
```

3 Convergence Study



(a) Mean value **(b)** Field integral

Figure 7. Mesh convergence of mean value and field integral

Attention is now turned to the mesh convergence of the solution to the QCAD non-linear Poisson problem posed on the Ottawa Flat 270 geometry discretized with a four-node tetrahedral mesh. Two metrics are used to study mesh convergence: the mean value of the solution, and the field integral of the solution (the integral of the solution over the domain). The former quantity is effectively an $L^1(\Omega)$ norm, and the latter is effectively an $L^2(\Omega)$ norm. From basic finite element theory, the expected convergence rates in these norms for four-node tetrahedral finite elements are one and two respectively [1]. Since an analytical form of the exact solution to this problem is not available, the relative errors were measured with respect to a converged reference solution, computed numerically on a mesh of ≈ 552 elements obtained by executing the `refine volume all numsplit 3` Cubit command on the coarsest (≈ 1 million element) mesh considered. Given u_{ref}, the computed solution on the reference mesh (in this case, the mean value of the solution or the field integral), the relative error was computed as:

$$\varepsilon_{\text{ref}} = \frac{|u_{\text{ref}} - u_N|}{|u_{\text{ref}}|}, \tag{2}$$

where u_N is the solution computed on a mesh of N tetrahedral elements. Subfigures (a) and (b) in Figure 7 illustrate respectively the convergence of the solution mean value and field integral with respect to successive mesh refinement. Although the solution appears to be converging in both metrics, the reader may observe that the convergence rates are below the rates expected from theory. Most likely, this can be attributed to an insufficiently accurate reference solution. A further convergence study, perhaps in a different metric, may be worthwhile to undertake in future validation studies.

4 Conclusions

The present document summarizes the results of a numerical preconditioner and convergence study for a QCAD non-linear Poisson problem posed on the Ottawa Flat 270 design geometry. This domain is discretized with four-node tetrahedral elements using the Cubit mesher tool. Finer meshes are generated by successively refining an initial "coarse" mesh. For the preconditioner study, nine `Ifpack` and three `ML` preconditioners are evaluated. Three sets of options for the `ML` default type preconditioners are considered. It is found that by selecting an `ML` preconditioner with repartitioning based on nodal coordinates using the `Zoltan` package, the total linear solve time can be improved relative to the previously-employed `Ifpack` preconditioner by a factor of more than two for problems posed on coarse as well as fine meshes. This option is therefore recommended for all problem sizes, and is now available in Albany thanks to some new development. A scalability plot generated for this problem illustrates that the total linear solve time does not scale optimally with the problem size and processor count. The reason for this is unclear at the present time and may be the subject of future work.

Following the preconditioner performance study, a convergence study with respect to successive mesh refinement is performed. The convergence is measured in two metrics, the mean value of the solution and the field integral. For the convergence study, a "reference" solution computed numerically on a very fine mesh is taken in place of an exact solution, as the exact solution to this problem is not available in closed analytic form. The convergence study suggests convergence in both metrics, albeit at rates slightly below the expected theoretical rates. It is recommended that future work include further convergence testing for validation purposes, perhaps in metrics other than the ones considered herein.

References

[1] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*. Springer-Verlag New York, Inc., New York, NY.

[2] L. Gaudreau, G. Granger, A. Kam, G. C. Aers, S. A. Studenikin, P. Zawadzki, M. Pioro-Ladriere, Z. R. Wasilewski, and A. S. Sachrajda. Coherent control and interplay of three-electron spin states in a triple quantum dot. *Submitted to Condensed Matter Physics*, 2011.

[3] M.W. Gee, C.M. Siefert, J.J. Hu, R.S. Tuminaro, and M.G. Sala. Ml 5.0 smoothed aggregation user's guide. Technical report, Sandia National Laboratories, 2007.

[4] E.P. Nordberg, G.A. Ten Eyck, H.L. Stalford, R.P. Muller, R.W. Young, K. Eng, L.A. Tracy, K.D. Childs, J.R. Wendt, R.K. Grubbs, J. Stevens, M.P. Lilly, M.A. Eriksson, and M. S. Carroll. Enhancement-mode double-top-gated metal-oxide-semiconductor nanostructures with tunable lateral geometry. *Phys. Rev. B 80*, 80(115331):1–10, 2009.

[5] M. Sala and M. Heroux. Robust algebraic preconditioners using ifpack 3.0. Technical report, Sandia National Laboratories, 2005.

DISTRIBUTION:

1	MS 1318	Andy Salinger, 1442
1	MS 1322	Rick Muller, 1425
1	MS 1415	Erik Nielsen, 1121
1	MS 1322	Suzey Gao, 1425
1	MS 9159	Ray Tuminaro, 1442
1	MS 9159	Jonathan Hu, 1426
1	MS 1322	Chris Siefert, 1443
1	MS 1084	Ralph Young, 1748
1	MS 1318	Erik Boman, 1426
1	MS 1318	Karen Devine, 1426
1	MS 1320	Scott Collis, 1442
1	MS 1318	Bruce Hendrickson, 1440
1	MS 1322	John Aidun, 1425
1	MS 1318	Rob Hoekstra, 1426
1	MS 0899	Technical Library, 9536 (electronic copy)

Sandia National Laboratories