A Comparison of Several Direct Sparse Linear Equation Solvers for CGWAVE on the Cray X1

Fred T. Tracy and Thomas C. Oppe, *U.S. Army Engineer Research and Development Center Major Shared Resource Center*

ABSTRACT: *A number of sparse direct linear equation solvers are compared for the solution of sets of linear equations arising from the wave climate analysis program CGWAVE developed at the Coastal and Hydraulics Laboratory at the U.S. Army Engineer Research and Development Center. CGWAVE is a general-purpose, state-of-the-art wave-prediction model. It is applicable for estimation of wave fields in harbors, open coastal regions, and coastal inlets, and for around islands and fixed or floating structures. CGWAVE generates systems of simultaneous, linear equations with complex coefficients that are difficult to solve by iterative methods. The vendor-supplied solvers SSGETRF/SSGETRS and SSTSTRF/SSTSTRS are compared with SuperLU, UMFPACK, and a direct, banded, out-of-core solver that utilizes reordering of the nodes to reduce the size of the bandwidth. The latter solver was optimized for the ERDC Major Shared Resource Center Cray X1.*

KEYWORDS: direct linear solver, wave model, Cray X1, SGI Origin 3900

1. CGWAVE

CGWAVE [1, 2] is a general-purpose, state-of-the-art wave-prediction model. It is applicable for estimation of wave fields in harbors, open coastal regions, and coastal inlets, and for around islands and fixed or floating structures. Thus, the results from CGWAVE can have significant military application. Figure 1 shows a generic computational domain. Both monochromatic and spectral waves can be simulated with the CGWAVE model. CGWAVE is a finite element model interfacing with the Corps of Engineers' Surface-Water Modeling System (SMS) [3] for graphics and efficient creation of finite element mesh generation and other input data.

CGWAVE was used to model harbors of military interest in various parts of the world to create a database for use by Department of Defense (DoD) organizations. This information is vital for planning and execution of military operations, training, and exercises.

Suitability

CGWAVE is particularly suited for performing wave simulations in regions with arbitrarily shaped (man-made or natural) boundaries and arbitrary depth variations. Intrinsic limitations do not exist on the shape of the domain,

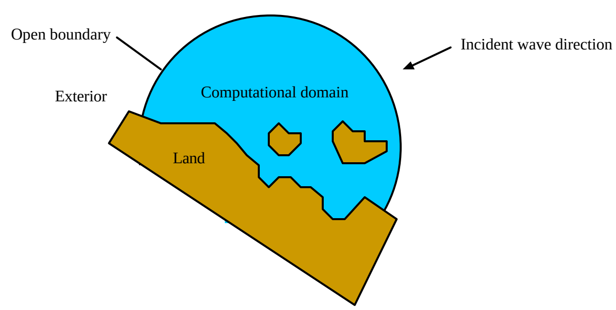

Figure 1. Computational domain

the angle of wave incidence, or the degree and direction of wave reflection and scattering that can be modeled. CGWAVE is one of the DoD's harbor-wave simulation models. It is used (a) in the planning and execution of operations in ports and harbors and (b) design/modifica-tion of commercial ports, marinas, and yacht basins. Engineers conducting studies on navigation, channel deepening, and fluid-structure interaction problems can also use CGWAVE.

Governing equations

CGWAVE is based on the solution of the elliptic mild-slope equation (MSE) for modeling surface-gravity waves in coastal areas. The MSE represents integration over water columns of the three-dimensional Laplace's equation used in the classical potential wave theory. The governing equation is as follows:

$$
\begin{aligned}
&\nabla \cdot \left(CC_g \hat{\eta}\right) + \frac{C}{C_g}\sigma^2 \hat{\eta} = 0 \\
&C = \frac{\sigma}{k} \\
&C_g = \frac{\partial \sigma}{\partial k} = nC \\
&n = \frac{1}{2}\left[1 + \frac{2kd}{\sinh\left(2kd\right)}\right] \\
&\sigma^2 = gk \tanh\left(kd\right)
\end{aligned}
\qquad (1)
$$

where $\hat{\eta}$ is the complex surface elevation function from which the wave height can be determined, s is the wave frequency under consideration (radians/sec), C is the phase velocity, C_g is the group velocity, k is the wave number, and d is the local water depth.

CGWAVE provides an estimate of the spatial variation in the wave field from the time incident waves enter the model domain in the deep ocean to the time they move through inlets/entrances and start impacting various maritime activities inside ports, harbors, and estuaries. The wave-field data model provides (a) wave height, direction, and speed, (b) pressure, and (c) wave-radiation stresses.

2. Linear solvers

Whether finite differences or finite elements are used for the discretization of the governing equations and boundary conditions of CGWAVE, the numerical treatment leads to a system of simultaneous, linear equations

$$\mathbf{Ax} = \mathbf{b} \qquad (2)$$

that must be solved. Here, $\mathbf{A}$ is a nonsymmetric, complex matrix of coefficients, $\mathbf{x}$ is the vector of unknown complex potentials, and $\mathbf{b}$ is a vector of known complex terms. The matrix $\mathbf{A}$ has a symmetric structure and is usually extremely large. Because this system of equations has challenged iterative solvers with varying success [4], direct solvers are investigated in this paper.

Direct linear solvers

Five direct linear solvers were tested with three CGWAVE data sets. Threshold pivoting values of 0.1 and 1.0 (classic partial pivoting) were used in the tests reported in this paper. Timings for the U.S. Army Engineer Research and Development Center (ERDC) Major Shared Resource Center (MSRC) Cray X1 and SGI Origin 3900 (O3K) were acquired whenever possible. All computations were accomplished using 64-bit arithmetic and, when possible, compiled with high optimization (e.g., using the –O3 flag). A description of each of the solvers is as follows:

1. **Name:** Bansol

   Source: ERDC MSRC

   Description: Out-of-core, banded solver. It uses an initial bandwidth reduction step. Complex coefficients are used. This solver was optimized using directives for the X1[5].

2. **Name:** SSGETRF, SSGETRS

   Source: Cray X1 SciLib library

   Description: General unsymmetric sparse solver. Phases include (a) fill-reduction reordering, (b) symbolic factorization, (c) execution sequence and memory management, (d) numerical factorization, and (e) back substitution. Real coefficients are used.

3. **Name:** SSTSTRF, SSTSTRS

   Source: Cray X1 SciLib library

   Description: Symmetric structure sparse solver. No pivoting is done. Phases include (a) fill-reduction reordering, (b) symbolic factorization, (c) execution sequence and memory management, (d) numerical factorization, and (e) back substitution. Real coefficients are used.

4. **Name:** SuperLU [6]

   Source: http://www.eecs.berkeley.edu/demmel/SuperLU.html

   Description: General unsymmetric sparse solver. Phases include (a) equilibrate $\mathbf{A}$, (b) preorder the rows of $\mathbf{A}$, (c) order the columns of $\mathbf{A}$, (d) compute the $\mathbf{LU}$ factorization of $\mathbf{A}$, and (e) apply back substitution. Real coefficients are used.

5. **Name:** UMFPACK [7]

   Source: http://www.cise.ufl.edu/research/sparse/umfpack

Description: General unsymmetric sparse solver. Uses the unsymmetric pattern multifrontal method. The symmetric approximate minimum degree ordering routine [8, 9] added. Phases include (a) preorder and symbolic analysis, (b), numerical factorization, and (c) back substitution.

In all but Bansol, real data are used, so the N complex equations are turned into a set of 2N real equations.

Test problems

Table 1 gives a description of the data sets from CGWAVE used in this study. One can note a small data set with a small bandwidth, a medium data set with a large bandwidth, and a large data set with a small bandwidth. The infinity norms of the solutions and right-hand sides vary a significant amount as well. These three problems are therefore a good representation of CGWAVE data.

Data Set ID	a	p11run24	event43
Nodes	130,255	265,119	496,286
Old half band-Width	719	1,487	1,829
New half band-Width	719	1,487	583
$\|\mathbf{x}_{bm}\|_\infty$	7.58	0.000443	2.92
$\|\mathbf{b}\|_\infty$	596.0	0.0288	108.0

Table 1. Data set properties

Bansol test results

Table 2 gives the timing and accuracy results for the Cray X1 for the banded out-of-core solver for threshold values of 0.1 and 1.0, and Table 3 gives the O3K results. The following key is used to conserve space in the table:

T_h	-	threshold value
BR	-	bandwidth reduction time
WDB	-	writing data in blocks time
LUC	-	**LU** factorization computation time
LUIO	-	**LU** factorization IO time
BSC	-	Back substitution computation time
BSIO	-	Back substitution IO time
Total	-	Total time for solution
$\mathbf{x}_{bm}$	-	Benchmark solution for comparison

Conclusions that can be drawn from these results are as follows:

1. The multistreaming directives added to the X1 version of the test code significantly reduced the run time.
2. Since the test data were derived from a finite element analysis, the **A** matrix is diagonally dominant. Thus, the full partial pivoting computation did not take much more time than the T_h = 0.1 run.

	T_h	a	p11run24	Event43
BR		0.7	1.6	3.2
WDB		73.7	317.6	246.1
LUC	0.1	189.1	1,416.1	415.8
	1.0	187.8	1,415.9	413.9
LUIO	0.1	141.7	762.9	581.5
	1.0	194.2	773.5	678.9
BSC	0.1	6.7	11.6	12.0
	1.0	6.7	10.9	12.2
BSIO	0.1	64.0	300.9	286.3
	1.0	79.6	311.6	316.2
Total	0.1	478.7	2,812.4	1,543.5
	1.0	538.7	2,826.9	1,665.6
$\|\mathbf{x}-\mathbf{x}_{bm}\|_\infty$	0.1	5.59 (10^{-13})	1.88 (10^{-15})	5.07 (10^{-11})
	1.0	5.60 (10^{-13})	1.90 (10^{-15})	5.07 (10^{-11})
$\|\mathbf{b}-\mathbf{Ax}\|_\infty$	0.1	1.41 (10^{-11})	1.25 (10^{-15})	1.13 (10^{-12})
	1.0	1.38 (10^{-11})	1.00 (10^{-15})	1.13 (10^{-12})

Table 2. Timings (sec) and accuracy for Bansol on the Cray X1

	T_h	a	p11run24	Event43
BR		0.1	0.3	0.5
WDB		79.7	348.2	379.8
LUC	0.1	1,316.0	19,130.7	2,264.7
	1.0	1,313.0	19,145.3	2,256.4
LUIO	0.1	141.7	2,134.5	702.0
	1.0	240.6	2,128.1	794.0
BSC	0.1	49.5	216.5	139.2
	1.0	48.7	211.9	141.1
BSIO	0.1	220.6	1,443.1	680.0
	1.0	221.8	1,761.8	716.7
Total	0.1	1,893.4	23,271.7	4,158.7
	1.0	1,909.9	23,596.6	4,295.2
$\|\mathbf{x}-\mathbf{x}_{bm}\|_\infty$	0.1	5.55 (10^{-13})	1.88 (10^{-15})	5.07 (10^{-11})
	1.0	5.58 (10^{-13})	1.90 (10^{-15})	5.07 (10^{-11})
$\|\mathbf{b}-\mathbf{Ax}\|_\infty$	0.1	1.80 (10^{-11})	1.52 (10^{-15})	1.20 (10^{-12})
	1.0	1.61 (10^{-11})	1.37 (10^{-15})	1.20 (10^{-12})

Table 3. Timings (sec) and accuracy for Bansol on the O3K

3. The full partial pivoting computation was only slightly more accurate than the T_h = 0.1 results as illustrated by the infinity norm of the residual.

4. The size of the bandwidth can affect the solution time more than the number of equations.

SSGETRF and SSGETRS test results

Table 4 gives the timing and accuracy results for the general unsymmetric sparse matrix solvers on the X1. The following key is used to conserve space in the table:

SUD	-	setting up data time
FRR	-	fill reduction reordering time
SF	-	symbolic factorization time
ESMM	-	execution sequence and memory management time
NF	-	numerical factorization time
BS	-	back substitution time
Total	-	Total time for solution

	T_h	a	p11run24	event43
SUD		30.6	41.7	303.6
FRR	0.1	18.1	33.2	65.5
	1.0	18.3	33.5	64.8
SF	0.1	3.2	6.4	12.2
	1.0	3.2	6.5	12.1
ESMM	0.1	0.06	0.11	0.21
	1.0	0.06	0.11	0.21
NF	0.1	18.7	65.4	142.4
	1.0	18.9	65.4	144.0
BS	0.1	2.9	5.8	10.9
	1.0	2.9	5.9	10.9
Total	0.1	73.4	152.2	534.0
	1.0	74.3	153.4	536.2
$\lVert x - x_{bm} \rVert_\infty$	0.1	11.1 (10^{-13})	1.92 (10^{-15})	5.07 (10^{-11})
	1.0	6.06 (10^{-13})	1.92 (10^{-15})	5.07 (10^{-11})
$\lVert b - Ax \rVert_\infty$	0.1	10.9 (10^{-11})	43.7 (10^{-16})	12.6 (10^{-11})
	1.0	1.49 (10^{-11})	3.86 (10^{-16})	1.46 (10^{-11})

Table 4. Timings (sec) and accuracy for SSGETRF and SSGETRS on the Cray X1

Conclusions that can be drawn from these results are as follows:

1. The sparse solver is much faster than the banded out-of-core solver.

2. As before, the full partial pivoting computation did not take much more time than the T_h = 0.1 run.

3. As before, the full partial pivoting computation was only slightly more accurate than the T_h = 0.1 run as illustrated by the infinity norm.

4. This solver is slightly less accurate than the banded solver.

SSTSTRF and SSTSTRS test results

Table 5 gives the timing and accuracy results for the symmetric structure sparse matrix solvers on the X1.

	A	p11run24	event43
SUD	30.1	41.5	305.0
FRR	14.3	25.2	51.1
SF	3.2	6.4	12.3
ESMM	0.03	0.06	0.12
NF	33.1	106.4	231.9
BS	2.7	5.4	10.3
Total	83.4	185.0	610.8
$\lVert x - x_{bm} \rVert_\infty$	1.32 (10^{-11})	1.92 (10^{-15})	5.13 (10^{-10})
$\lVert b - Ax \rVert_\infty$	3.82 (10^{-9})	1.73 (10^{-14})	2.13 (10^{-8})

Table 5. Timings (sec) and accuracy for SSTSTRF and SSTSTRS on the Cray X1

Conclusions that can be drawn from these results are as follows:

1. Surprisingly, the timings for this solver were greater than for the general solver.

2. The accuracy was slightly worse as a result of no pivoting, but the results are still acceptable.

SuperLU test results

Table 6 gives the timing and accuracy results for the SuperLU solver on the X1, and Table 7 gives the results for the O3K. No optimization was done to the library. The following key is used to conserve space in the table:

F	-	factorization time
BS	-	back substitution time
Total	-	Total time for solution

Conclusions that can be drawn from these results are as follows:

1. The X1 results are worse than the banded solver for the data set event43 and significantly worse than the library solvers. This emphasizes the need to optimize the code. However, for the data set p11run24, the size of the bandwidth is such that Bansol is very slow.

2. A comparison of Tables 3 and 7 shows that the SuperLU sparse solver performs better on the O3K than Bandsol.

3. As before, the full partial pivoting computation did not take much more time than the $T_h = 0.1$ run.

4. As before, the full partial pivoting computation was only slightly more accurate than the $T_h = 0.1$ run as illustrated by the infinity norm.

	T_h	a	p11run24	event43
SUD		30.1	41.0	305.9
F	0.1	199.4	633.2	1,322.4
	1.0	199.3	636.3	1,348.9
BS	0.1	13.2	36.3	65.4
	1.0	13.2	36.6	67.0
Total	0.1	242.7	710.5	1,690.8
	1.0	242.6	714.0	1,724.8
$\|x - x_{bm}\|_\infty$	0.1	11.1 (10^{-13})	1.91 (10^{-15})	5.06 (10^{-11})
	1.0	5.97 (10^{-13})	1.91 (10^{-15})	5.07 (10^{-11})
$\|b - Ax\|_\infty$	0.1	2.66 (10^{-10})	45.3 (10^{-16})	51.0 (10^{-11})
	1.0	1.59 (10^{-10})	7.43 (10^{-16})	5.42 (10^{-11})

Table 6. Timings (sec) and accuracy for SuperLU on the Cray X1

	T_h	a	p11run24	event43
SUD		1.8	2.9	19.1
F	0.1	93.8	603.4	1,706.7
	1.0	95.5	628.2	1,791.0
BS	0.1	2.2	6.1	443.3
	1.0	2.5	7.1	427.2
Total	0.1	97.8	612.6	2,171.5
	1.0	99.9	637.9	2,236.1
$\|x - x_{bm}\|_\infty$	0.1	16.4 (10^{-13})	1.90 (10^{-15})	5.08 (10^{-11})
	1.0	5.88 (10^{-13})	1.91 (10^{-15})	5.07 (10^{-11})
$\|b - Ax\|_\infty$	0.1	3.56 (10^{-10})	54.7 (10^{-16})	5.51 (10^{-11})
	1.0	1.50 (10^{-10})	7.09 (10^{-16})	4.61 (10^{-11})

Table 7. Timings (sec) and accuracy for SuperLU on the O3K

UMFPACK test results

Table 8 gives the timing and accuracy results for UMFPACK on the X1, and Table 9 gives the results for the O3K. No optimization was done to the library. The following key is used to conserve space in the table:

POSA - preorder and symbolic analysis time
NF - numeric factorization time
BS - back substitution time
Total - Total time for solution

	T_h	a	p11run24	event43
SUD		30.4	42.1	305.2
POSA	0.1	47.7	143.9	294.6
	1.0	48.0	143.0	295.0
NF	0.1	71.0	240.5	496.5
	1.0	70.6	232.2	668.2
BS	0.1	1.7	4.2	9.1
	1.0	1.7	4.1	10.3
Total	0.1	151.0	431.2	1,105.9
	1.0	150.6	421.0	1,277.9
$\|x - x_{bm}\|_\infty$	0.1	1.81 (10^{-12})	1.90 (10^{-15})	5.05 (10^{-11})
	1.0	1.85 (10^{-12})	1.90 (10^{-15})	5.09 (10^{-11})
$\|b - Ax\|_\infty$	0.1	18.5 (10^{-10})	5.06 (10^{-15})	54.3 (10^{-11})
	1.0	7.38 (10^{-10})	3.37 (10^{-15})	3.40 (10^{-11})

Table 8. Timings (sec) and accuracy for UMFPACK on the Cray X1

	T_h	a	p11run24	event43
SUD		0.7	1.2	4.7
POSA	0.1	4.9	13.6	28.6
	1.0	5.0	13.6	28.6
NF	0.1	27.5	146.3	325.8
	1.0	27.5	146.7	528.6
BS	0.1	0.7	1.9	4.0
	1.0	0.7	1.9	4.4
Total	0.1	33.8	163.0	363.1
	1.0	33.8	163.4	566.6
$\|x - x_{bm}\|_\infty$	0.1	10.3 (10^{-12})	1.93 (10^{-15})	5.04 (10^{-11})
	1.0	4.46 (10^{-12})	1.93 (10^{-15})	27.6 (10^{-11})
$\|b - Ax\|_\infty$	0.1	3.20 (10^{-9})	1.91 (10^{-14})	1.47 (10^{-9})
	1.0	1.36 (10^{-9})	1.38 (10^{-14})	1.82 (10^{-9})

Table 9. Timings (sec) and accuracy for UMFPACK on the O3K

Conclusions that can be drawn from these results are as follows:

1. UMFPACK outperforms SuperLU.

2. The O3K outperforms the X1 at times because of the lack of optimization of the solver library on the X1.

3. The full partial pivoting computation took considerably more time than the $T_h = 0.1$ run for the large event43 data set.

4. As before, the full partial pivoting computation was only slightly more accurate than the $T_h = 0.1$ run as illustrated by the infinity norm of the residual.

3. Overall conclusions

Figure 2 shows a plot of running times for all five of the linear solvers tested for the three example problems for $T_h = 1.0$, and Figure 3 shows the O3K run times for the three nonproprietary solvers. Overall, the best solver for the X1 is the general unsymmetric system library solver SSGETRF/SSGETRS, and the best solver for the O3K is UMFPACK.

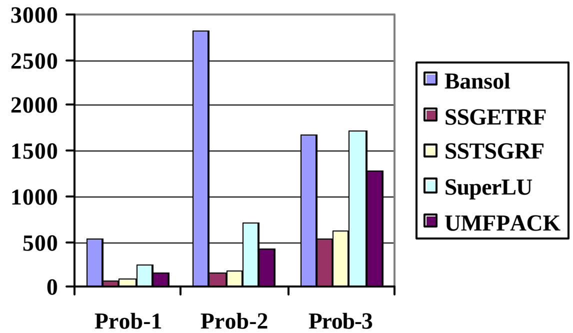

Figure 2. Comparison of the solvers on the Cray X1

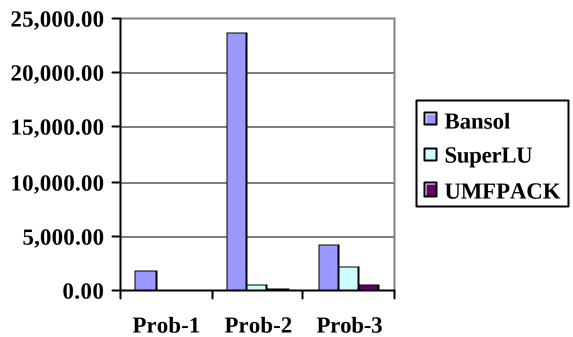

Figure 3. Comparison of the solvers on the O3K

Acknowledgment

This work was supported in part by a grant of computer time from the DoD High Performance Computing Modernization Program at the ERDC MSRC, Information Technology Laboratory, Vicksburg, MS.

References

1. Z. Demirbilek, CGWAVE (computer program), http://chl.wes.army.mil/research/wave/wavesprg/numeric/wentrances/cgwave.htp, U.S. Army Engineer Research and Development Center (ERDC), Vicksburg, MS, 2004.

2. Z. Demirbilek and V. Panchang, "CGWAVE: A Coastal Surface Water Wave Model of the Mild Slope Equation," Technical Report, U.S. Army Engineer Research and Development Center (ERDC), Vicksburg, MS, 1998.

3. GMS Team, GMS (computer program), http://chl.erdc.usace.army.mil/CHL.aspx?p=s&a=Articles;26, U.S. Army Engineer Research and Development Center (ERDC), Vicksburg, MS, 2005.

4. J. Tang, Y. Shen, Y. Zheing, and D. Qiu, "An Efficient and Flexible Computational Model for Solving the Mild Slope Equation," Coastal Engineering, Elsevier, pp. 143-154, 2004.

5. F. Tracy, T. Oppe, and Z. Demirbilek, "A Direct Out-of-Core solver for CGWAVE on the Cray X1," in Proc. DoD HPCUG, pp. 228-232, 2004.

6. J. Demmel, J. Gilbert, and X. Li, "SuperLU Users' Guide," University of California, Berkeley and Santa Barbara, CA, 2003.

7. T. Davis, "UMFPACK Version 4.4 User Guide," Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, 2005.

8. P. Amestoy, T. Davis, and I. Duff, "An Approximate Minimum Degree Ordering Algorithm," SIAM J. Matrix Anal. Applic., 17(4), pp. 886–905, 1996.

9. P. Amestoy, T. Davis, and I. Duff, "Algorithm 837: AMD, An Approximate Minimum Degree Ordering Algorithm," ACM Trans. Math. Softw., 30(3), pp. 381–388, 2004.