

11-39-CR
27814

COMPARISON OF SYMBOLIC AND NUMERICAL INTEGRATION METHODS FOR AN ASSUMED-STRESS HYBRID SHELL ELEMENT

Govind Rengarajan*

Texas A&M University, College Station, TX 77843-3123

Norman F. Knight, Jr.† and Mohammad A. Aminpour‡

Old Dominion University, Norfolk, VA 23529-0247

Abstract

Hybrid shell elements have long been regarded with reserve by the commercial finite element developers despite the high degree of reliability and accuracy associated with such formulations. The fundamental reason is the inherent higher computational cost of the hybrid approach as compared to the displacement-based formulations. However, a noteworthy factor in favor of hybrid elements is that numerical integration to generate element matrices can entirely be avoided by the use of symbolic integration. In this paper, the use of the symbolic computational approach is presented for an assumed-stress hybrid shell element with drilling degrees of freedom and the significant time savings achieved is demonstrated through an example.

* Graduate Research Assistant, Department of Mechanical Engineering, Formerly at Department of Mechanical Engineering, Clemson University, Clemson, SC 29634-0921.

† Associate Professor, Department of Aerospace Engineering.

‡ Associate Professor, Department of Aerospace Engineering. Formerly at Analytical Services and Materials, Inc., Hampton, VA 23666

(NASA-CR-195945) COMPARISON OF
SYMBOLIC AND NUMERICAL INTEGRATION
METHODS FOR AN ASSUMED-STRESS
HYBRID SHELL ELEMENT (Texas A&M
Univ.) 15 p

N95-13096

Unclass

G3/39 0027814

Introduction

Evaluation of the finite element quantities (i.e., stiffness and mass matrices, load vector, etc.) is usually performed using numerical integration with an appropriate Gaussian quadrature rule (e.g., see [1]). However, two significant issues arise. First, if the integrand is a polynomial then an appropriate order Gaussian quadrature rule will result in exact integration. Second, as the complexity of element formulation increases so does the computational effort to evaluate element matrices and vectors. In the standard displacement formulation for parallelogram-shaped elements, the order of the Gaussian quadrature rule can be specified *a priori*. However for general quadrilateral-shaped (i.e., distorted) elements, the Jacobian is dependent on the element natural coordinates, and hence, the integrand is a ratio of polynomials. As such, numerical integration with a given Gaussian quadrature rule results in an approximation.

In finite element formulations based on multi-field variational principles (mixed formulations), the element matrices could be generated explicitly such that element operations like numerical integration are entirely avoided. This is achieved by using symbolic computational techniques and symbolic manipulation software (e.g., see MACSYMA [2], MAPLE [3], MATHEMATICA [4]).

The applicability of these computational techniques is by no means restricted to finite element analysis or computational mechanics alone (e.g., see Beltzer [5] and Tsai and Kikuchi [6]); however, as this paper deals with a successful implementation of a symbolic computational approach in a finite element code, in what follows we shall focus primarily on symbolic methods within the context of a finite element method. Symbolic and algebraic manipulation software systems for the evaluation of finite element arrays has been used for approximately two decades beginning with the early work of Gunderson and Cetiner [7] and Luft et al. [8]. Andersen and Noor [9], Korncoff [10], Korncoff and Fenves [11], Noor and Andersen [12, 13], and Tan et al. [14] give an account of the development of symbolic manipulation in finite element code

generation. In each case the symbolic approach is noted for its robustness, accuracy, reliability, and for assisting in algebraic manipulations involved in research towards new finite element formulations. While most studies concentrated on displacement-based formulations, several researchers have examined mixed and hybrid formulations (e.g., Noor and Andersen [12], Saleeb and Chang [15], Chang et al. [16], and Tan et al. [14]). These studies focussed primarily on the symbolic computational aspects of the problem and present very limited data, if any, comparing the computational effort required for the symbolic and numerical approaches. Performance data comparing other shell element implementations on a nearly level playing field in terms of computational effort to achieve a specific solution error have not been presented.

The objective of this paper is to present comparison demonstrating the effectiveness of a symbolic computational approach over the numerical integration approach for a 4-node assumed-stress hybrid shell element with drilling degrees of freedom. In general, hybrid formulations involve more computations at the element level than displacement-based formulations and hence are often shunned by developers of commercial finite element software systems. However, hybrid formulations readily lend themselves to symbolic computational techniques even for distorted elements. The symbolic approach offers significant computational savings over the traditional numerical integration approach and thereby should make these elements attractive from both a computation and accuracy point of view. A brief description is first given for the hybrid element formulation and finite element approximations, followed by a description of the use of the symbolic computational strategy. Finally, the effectiveness of the present strategy is demonstrated through a timing-study on a specific example.

Hybrid Element Formulation

The hybrid element formulation used in this paper is based on the Hellinger-Reissner multi-field variational principle and is given in detail in reference [17]. A brief outline of the formulation is given herein for completeness and to clarify the

notation used herein. The ensuing discussion pertains to a solid elastic continuum V , with boundary S . Let S_σ be the section of the boundary where tractions are prescribed; let S_u be the section of the boundary where displacements are prescribed. The Hellinger-Reissner functional can be written as

$$\Pi_{HR} = -\frac{1}{2} \int_V \{\sigma\}^T [D] \{\sigma\} dV + \int_V \{\sigma\}^T [\mathcal{L}] \{u\} dV - \int_{S_\sigma} \{u\}^T \{t_o\} dS, \quad (1)$$

where $[D]$ is the compliance matrix, $[\mathcal{L}]$ is the linear differential operator on the displacements $\{u\}$ to produce strains, and $\{t_o\}$ is the prescribed tractions on S_σ .

The approximations for stresses and displacements can now be incorporated in the functional. The stress field is described in the interior of the element as

$$\{\sigma\} = [P]\{\beta\}, \quad (2)$$

and a compatible displacement field is described by

$$\{u\} = [N]\{q\}, \quad (3)$$

where $[P]$ and $[N]$ are matrices of stress and displacement interpolation functions, respectively, and $\{\beta\}$ and $\{q\}$ are the unknown stress and nodal displacement parameters, respectively. Substituting the stress and displacement approximations [equations (2) and (3)] in the functional Π_{HR} [equation (1)] results in

$$\Pi_{HR} = -\frac{1}{2} \{\beta\}^T [H] \{\beta\} + \{\beta\}^T [T] \{q\} - \{q\}^T \{F\}, \quad (4)$$

where

$$\begin{aligned} [H] &= \int_V [P]^T [D] [P] dV, \\ [T] &= \int_V [P]^T [\mathcal{L}] [N] dV, \\ \{F\} &= \int_{S_\sigma} [N]^T \{t_o\} dS. \end{aligned} \quad (5)$$

Imposing stationary conditions on the functional with respect to the stress parameters $\{\beta\}$ gives

$$\{\beta\} = [H]^{-1} [T] \{q\}. \quad (6)$$

Upon substituting equation (6) into equation (4), the functional reduces to

$$\Pi_{HR} = \frac{1}{2}\{q\}^T[K]\{q\} - \{q\}^T\{F\}, \quad (7)$$

where $[K]$, the elemental stiffness matrix, is given by

$$[K] = [T]^T[H]^{-1}[T]. \quad (8)$$

Imposing the stationary conditions on the functional with respect to nodal displacement unknowns $\{q\}$ results in a system of equations for the element of the form

$$[K]\{q\} = \{F\}. \quad (9)$$

Assembly of the element equations for a given finite element model and application of boundary conditions then defines the global system of linear algebraic equations to be solved.

Displacement Field Approximations

The generalized displacements and rotations represented in terms of interpolation functions and nodal values of the displacements and rotations are

$$\begin{aligned} u^o(\xi, \eta) &= \sum_{i=1}^4 N_i u_i + \sum_{i=1}^4 \frac{\Delta y_i}{8} N_i^* (\theta_{xj} - \theta_{xi}), \\ v^o(\xi, \eta) &= \sum_{i=1}^4 N_i v_i - \sum_{i=1}^4 \frac{\Delta x_i}{8} N_i^* (\theta_{xj} - \theta_{xi}), \\ w^o(\xi, \eta) &= \sum_{i=1}^4 N_i w_i - \sum_{i=1}^4 \frac{\Delta y_i}{8} N_i^* (\theta_{xj} - \theta_{xi}) + \sum_{i=1}^4 \frac{\Delta x_i}{8} N_i^* (\theta_{yj} - \theta_{yi}), \\ \theta_x(\xi, \eta) &= \sum_{i=1}^4 N_i \theta_{xi}, \\ \theta_y(\xi, \eta) &= \sum_{i=1}^4 N_i \theta_{yi}, \end{aligned} \quad (10)$$

where

$$N_i = \frac{1}{4}(1 + \xi_i \xi)(1 + \eta_i \eta) ; i = 1, 2, 3, 4 \quad (11)$$

$$N_i^* = \begin{cases} \frac{1}{2}(1 - \xi^2)(1 + \eta_i \eta) & ; i = 1, 3 \\ \frac{1}{2}(1 - \eta^2)(1 + \xi_i \xi) & ; i = 2, 4 \end{cases} \quad (12)$$

and

$$j = \begin{cases} i+1 & ; i = 1, 2, 3 \\ 1 & ; i = 4 \end{cases} \quad (13)$$

In equations (10)–(13), ξ and η denote the element natural coordinates, and ξ_i and η_i denote the values of ξ and η at node i . Note that the inplane displacement field approximations include contributions from the drilling degrees of freedom in order to increase the level of approximation over the element without introducing additional nodes in the element. Further details on the displacement field and shape functions are given in references [17, 18].

Stress Field Approximations

The stress resultant field for the membrane part is assumed to be

$$\begin{aligned} N_\xi &= \beta_1 + \beta_4\xi + \beta_6\eta + \beta_8\eta^2, \\ N_\eta &= \beta_2 + \beta_5\xi + \beta_7\eta + \beta_9\xi^2, \\ N_{\xi\eta} &= \beta_3 - \beta_4\eta - \beta_7\xi. \end{aligned} \quad (14)$$

The stress resultant field approximation for the bending part is assumed to be

$$\begin{aligned} M_\xi &= \bar{\beta}_1 + \bar{\beta}_4\xi + \bar{\beta}_6\eta + \bar{\beta}_8\eta^2, \\ M_\eta &= \bar{\beta}_2 + \bar{\beta}_5\xi + \bar{\beta}_7\eta + \bar{\beta}_9\xi^2, \\ M_{\xi\eta} &= \bar{\beta}_3 + \bar{\beta}_{10}\xi + \bar{\beta}_{11}\eta + \frac{1}{2}\bar{\beta}_{12}\xi^2 + \frac{1}{2}\bar{\beta}_{13}\eta^2. \end{aligned} \quad (15)$$

The transverse shear stress resultants are obtained by relating the stress parameters of the transverse shear stress resultants to those of the bending stress resultants, as

$$\begin{aligned} Q_\xi &= \bar{\beta}_4 + \bar{\beta}_{11} + \bar{\beta}_{13}\eta, \\ Q_\eta &= \bar{\beta}_7 + \bar{\beta}_{10} + \bar{\beta}_{12}\xi. \end{aligned} \quad (16)$$

Note that while the membrane and bending stress resultant field approximations are uncoupled, the transverse shear stress resultant field is coupled to the bending stress

resultant field, and the equilibrium equations are satisfied *a priori* for regular (rectangular) shapes. The stress field approximations are expressed in the element natural coordinate basis, and the stress interpolation function matrix $[P]$ is not uncoupled. This makes it virtually impossible to employ partitioning techniques in the evaluation of $[H]$ and $[T]$ matrices as was done in reference [14].

Symbolic Computation Approach

The evaluation of elemental stiffness matrix $[K]$, equation (8), requires the evaluation of $[H]$ and $[T]$ matrices [equation (5)]. The $[H]$ matrix is rewritten in a convenient form as

$$[H] = \int_{-1}^1 \int_{-1}^1 [\bar{H}(\xi, \eta)] d\xi d\eta \quad (17)$$

where

$$[\bar{H}(\xi, \eta)] = [P]^T [D] [P] \det|J|. \quad (18)$$

For this shell element, the matrix $[H]$ is a 22×22 matrix.

Similarly the $[T]$ matrix is expressed as

$$[T] = \int_{-1}^1 \int_{-1}^1 [\bar{T}(\xi, \eta)] d\xi d\eta \quad (19)$$

where

$$[\bar{T}(\xi, \eta)] = [P]^T [\mathcal{L}] [N] \det|J|. \quad (20)$$

For this shell element, the matrix $[T]$ is a 22×24 matrix.

In the symbolic evaluation of the $[H]$ and $[T]$ matrices, the integration is performed exactly and no approximation is introduced. Moreover the need to perform the integration for every element during an actual analysis is overcome – only the evaluation of symbolically generated results is required. Note that in equation (20), the differential operator $[\mathcal{L}]$ acting on $[N]$ implicitly introduces a $\det|J|$ in the denominator which cancels the $\det|J|$ appearing in the numerator. This is typical of the hybrid approach and makes it ideal for symbolic evaluation of elemental stiffness matrix. However in displacement-based elements, a $\det|J|$ will remain in the denominator

and hence will make direct symbolic integration very cumbersome or impractical, at least with the available software today, unless the element is restricted to a parallelogram shape with evenly spaced nodes and straight edges. One approach proposed by Andersen and Noor [9] to alleviate this inherent difficulty is to combine the use of symbolic and numerical integration methods.

In the numerical integration version, a 3×3 Gaussian rule is used to integrate these matrices exactly. Symbolic evaluation of the matrix inverse of $[H]$ for this shell element leads to symbolic expressions that are too cumbersome for most FORTRAN compilers. This occurs even when the matrix inverse is represented by independent expressions for its determinant and cofactor matrix. Instead, the term $[A] = [H]^{-1}[T]$ is evaluated numerically by solving the following system of linear algebraic equations which has multiple right-hand-side vectors (i.e., columns of the matrix $[T]$), or

$$[H][A] = [T] \quad (21)$$

Hence the explicit matrix inversion is avoided and efficient, optimized equation solvers can be exploited.

Results

To demonstrate the effectiveness of the symbolic integration approach over the numerical integration approach, a timing study is presented. For convenience, the numerical integration version of the present element formulation is referred to as A4N1 and the symbolic integration version as A4S1. These elements are implemented in the COmputational MEchanics Testbed COMET using the generic element processor [19]. As such, a common basis for the evaluation of different element formulations and implementations is readily and systematically accomplished. Common utilities for model generation, assembly and solution are available, thereby allowing element developers to focus only on element related issues. The other shell elements available in COMET include a 4-node incompatible assumed natural-coordinate strain element (4_ANS), a 4-node incompatible displacement-based element with drilling degrees of

freedom (4.STG), and a 4-node hybrid shell element without drilling degrees of freedom (4.HYB). Note that the element routines for both 4.ANS and 4.STG have been highly optimized within COMET, whereas the element routines for both A4N1 and A4S1 are research-oriented implementations. With the exception of the element routines, all other aspects of the analysis (e.g., data handling, element assembly, equation solving, etc.) are identical in that COMET provides a level playing field for assessing element performance.

The emphasis here is to compare the use of symbolic and numerical integration procedures and to determine their effect on computation time for evaluating element matrices and vectors. As such, only a single example problem is considered. For this purpose, an isotropic square plate with all edges clamped and subjected to a central concentrated load is considered (see Figure 1 and reference [17]). Due to symmetry, finite element models of only one-quarter of the plate are used in the computations. The mesh is refined and the CPU time to form the stiffness matrix is calculated at each level of mesh refinement.

The average computational effort to evaluate the element stiffness matrix as a function of the number of elements in the finite element model is shown in Figure 1. The CPU time per element converges to a constant value in all cases after approximately 30 elements. These results indicate that the 4.ANS and 4.STG elements require approximately the same computational effort and are the fastest of the element implementations considered herein. Also note that the symbolic integration approach (A4S1) requires only half the computational effort of the numerical integration approach (A4N1). In effect, the symbolic integration approach offers a gain of almost 50% computational time over the numerical integration approach. It is anticipated that another 10-20% reduction in CPU time per element is achievable once the element routines are optimized. Currently A4S1 requires just under twice the CPU time per element as either 4.ANS or 4.STG to evaluate the linear stiffness matrix.

A performance comparison should also take into account the accuracy of the predicted solution by different elements. It is possible to define a weighted average performance measure that takes into account both the CPU time and the error in the solution. The real difficulty lies in assigning appropriate weights for CPU time and solution error, since these depend on the user requirements. Hence in the present study, a more general representation is adopted by comparing the CPU time with the solution error. Solution error is defined for this problem as the absolute value of the ratio of the difference between the value of the center transverse deflection from classical plate theory and the value from the finite element analysis to the classical plate theory value, in percent.

The variation of the solution error as number of elements increases is shown in Figure 2. It is evident from Figure 2 that the present element formulation implemented as either A4N1 or A4S1 offers the best solution in that it converges to the exact value faster than the other elements considered. Using the results given in Figures 1 and 2, the CPU time to form all the element stiffness matrices of the finite element mesh may be estimated. Table I gives an indication of the computational effort to achieve a specified solution error. The total CPU time to evaluate all the element stiffness matrices (t^K) and the overall solution time (t^O) for each element type considered are normalized with respect to the corresponding times obtained using the A4S1 element and are presented in the last two columns of Table I. For a specified solution error percentage, 4_ANS takes the least time. Among the other elements, A4S1 offers the best time. In fact, 4_STG and 4_HYB do not predict a solution error of less than 0.5 % even with 144 elements. Though 4_ANS uses a larger number of elements to obtain a specified solution error than A4S1, it offers a better CPU time since it requires only half the CPU time per element compared with A4S1. However, both 4_ANS and 4_STG have been highly optimized for performance in terms of CPU time in COMET, while the element routines for A4S1 are still in a development form.

In general, as the robustness of an element formulation increases, the computational effort required to evaluate element properties also usually increases. For these hybrid elements, symbolic computations significantly reduce the CPU time required to evaluate the element stiffness matrix (compare results for A4S1 and A4N1). Additional improvements in the computing times are possible by reviewing the symbolically generated FORTRAN code and making any necessary enhancements to reduce the CPU time per element. Furthermore, an optimization study for A4S1 similar to 4_ANS and 4_STG is expected to yield additional computational time savings. With these enhancements, it is anticipated that the A4S1 element will exhibit total CPU times comparable to the 4_ANS element and perhaps better.

Furthermore, it is worth noting that both the 4_ANS and 4_STG elements are more sensitive to mesh distortion than the present A4S1 hybrid element as shown in reference [17]. While the performance data presented herein is for a uniform mesh of square elements, the computational effort to achieve a specific solution error with a mesh of distorted elements will increase significantly for the 4_ANS and 4_STG element cases. In these cases, the number of these elements required to achieve a given accuracy will be much larger than that required for the A4S1 element. Therefore, it is anticipated that for distorted meshes, the A4S1 hybrid element will offer improved performance over the 4_ANS element.

Conclusions

The use of symbolic computational approach in a 4-node hybrid shell element is presented and is shown to be almost twice as fast as the numerical version of the same element. This gain of computational time is significant as it makes the hybrid element, hitherto known to be computationally costly despite being more accurate, compete with efficient 4-node displacement-based elements.

On the symbolic approach front, the MAPLE output expressions are rather unwieldy; however, the expressions are automatically converted to FORTRAN statements and hence could be directly incorporated in to an existing finite element code.

Acknowledgment

The research reported herein was sponsored by NASA Langley Research Center and Dr. Alexander Tessler and Mr. Keith Norwood are the technical monitors. The first and second authors were sponsored by NASA Grant NAG-1-1374, and the third author was sponsored by NASA Contract NAS1-19317 while at AS&M.

References

- [1] Cook, R.D., Malkus, D.S., and Plesha, M.E., *Concepts and Applications of Finite Element Analysis*, Third Edition, John Wiley and Sons Inc., New York, 1989.
- [2] Pavelle, R., *MACSYMA: Capacities and Applications to Problems in Engineering and the Sciences*, Symbolics Inc., Cambridge, MA, 1985.
- [3] Char, B.W., Geddes, K.O., Gonnet, G.H., Leong, B.L., Monagan, M.B., and Watt, S.M., *MAPLE V Library Reference Manual*, Springer-Verlag, New York, 1991.
- [4] Wolfram, S., *MATHEMATICA - A System for Doing Mathematics by Computer*, Addison-Wesley, 1988.
- [5] Beltzer, A.I., *Variational and Finite Element Methods - A Symbolic Computational Approach*, Springer-Verlag, Berlin, 1990.
- [6] Tsai, W-L., and Kikuchi, N., "Application of Symbolic Algebraic Manipulation Software in Solving Applied Mechanics Problems", *NASA CR-4544*, August 1993.
- [7] Gunderson, R.H., and Cetiner, A., "Element Stiffness Matrix Generator", *Journal of the Structural Division, ASCE*, Vol. 97, No. ST1, 1971, pp. 363-375.
- [8] Luft, W., Roesset, J.M., and Conners, J.J., "Automatic Generation of Finite Element Matrices", *Journal of the Structural Division, ASCE*, Vol. 97, No. ST1, 1971, pp. 349-363.
- [9] Andersen, C.M., and Noor, A.K., "A Computerized Symbolic Integration Technique for Development of Triangular and Quadrilateral Composite Shallow-Shell Finite Elements", *NASA TN D-8067*, 1975.
- [10] Korncoff, A.R., "Finite Element Stiffness Matrix Generation via Symbolic Manipulation", *Proceedings of the Symposium on Application of Computer Methods in Engineering - Vol. II*, L. Carter Wellford, Jr. (editor), August 23-26, 1977, pp. 779-785.

- [11] Korncoff, A.R., and Fenves, S.J., "Symbolic Generation of Finite Element Stiffness Matrices", *Computers and Structures*, Vol. 10, 1979, pp. 119-124.
- [12] Noor, A.K., and Andersen, C.M., "Computerized Symbolic Manipulation in Structural Mechanics - Progress and Potential", *Computers and Structures*, Vol. 10, 1979, pp. 95-118.
- [13] Noor, A.K., and Andersen, C.M., "Computerized Symbolic Manipulation in Non-linear Finite Element Analysis", *Computers and Structures*, Vol. 13, 1981, pp. 379-403.
- [14] Tan, H.Q., Chang, T.Y.P., and Zheng, D., "On Symbolic Manipulation and Code Generation of a Hybrid Three-dimensional Solid Element", *Engineering with Computers*, Vol. 7, 1991, pp. 47-59.
- [15] Saleeb, A.F., and Chang, T.Y., "An Efficient Quadrilateral Element for Plate Bending Analysis", *International Journal for Numerical Methods in Engineering*, Vol. 24, 1987, pp. 1123-1155.
- [16] Chang, T.Y., Tan, H.Q., Zheng, D., and Yuan, M.W., "Application of Symbolic Method to Hybrid/Mixed Finite Elements and Computer Implementation", *Computers and Structures*, Vol. 35, No. 4, 1990, pp. 293-299.
- [17] Rengarajan, G., Aminpour, M.A., and Knight, N.F., Jr., "Improved Assumed-Stress Hybrid Shell Element with Drilling Degrees of Freedom for Linear Stress, Buckling, and Free Vibration Analyses", Submitted to the *International Journal for Numerical Methods in Engineering* for Review.
- [18] Aminpour, M.A., "An Assumed-Stress Hybrid 4-Node Shell Element with Drilling Degrees of Freedom", *International Journal for Numerical Methods in Engineering*, Vol. 33, 1992, pp. 19-38.
- [19] Stanley, G.M., and Nour-Omid, S., "The Computational Structural Mechanics Testbed Generic Structural Element Processor Manual", *NASA CR-181728*, 1990.

Table I. Computational effort required to achieve a specified solution error.

Solution Error	Element Name	Number of elements	t^K/t_{A4S1}^K	t^O/t_{A4S1}^O
< 5 %	A4S1	9	1.000	1.000
	A4N1	9	1.450	1.044
	4_ANS	16	0.763	0.835
	4_STG	16	0.919	1.098
	4_HYB	16	1.850	1.146
< 2 %	A4S1	16	1.000	1.000
	A4N1	16	1.650	1.114
	4_ANS	25	0.793	0.883
	4_STG	36	1.147	1.375
	4_HYB	49	3.622	1.738
< 1 %	A4S1	36	1.000	1.000
	A4N1	36	1.783	1.198
	4_ANS	49	0.758	0.891
	4_STG	100	1.440	1.978
	4_HYB	121	4.486	2.499
< 0.5 %	A4S1	64	1.000	1.000
	A4N1	64	1.809	1.269
	4_ANS	81	0.706	0.884
	4_STG	-	-	-
	4_HYB	-	-	-

t^K : Total CPU time to evaluate all the element stiffness matrices.

t^O : Overall solution time (total time from start to finish).

t_{A4S1}^K : Total CPU time to evaluate all the element stiffness matrices using A4S1.

t_{A4S1}^O : Overall solution time using A4S1.