

A Modified Non-Linear Transformation Method for Evaluating Weakly Singular Boundary Integrals

Barbara M. Johnston and Peter R. Johnston*

School of Science

Griffith University

Kessels Rd

Nathan

Queensland

Australia 4111

Phone: 61-7-3875-7748

Fax: 61-7-3875-7656

E-mail: Barbara.Johnston@mailbox.gu.edu.au

November 18, 2002

*Corresponding author. P.Johnston@mailbox.gu.edu.au

Abstract

Accurate numerical evaluation of boundary integrals is fundamental to producing useful results with the boundary element method. This paper introduces a generalisation of a recently introduced combined method (subtraction of singularity followed by a non-linear transformation), which takes into account the effect of the basis functions. The new method is applied to solve weakly singular integrals which arise in the solution of the two dimensional Laplace equation. The new method was found, in the cases considered, to be numerically superior to both the combined method and any of the non-linear transformation methods.

Running Title: Evaluating Weakly Singular Integrals.

Keywords: Singular Integrals, Boundary Element Method, Non-Linear Transformation, Singularity Subtraction, Laplace's Equation.

1 Introduction

When the boundary element method is used to solve the two dimensional Laplace equation, it is necessary to be able to accurately numerically integrate the weakly singular boundary integrals involved. Considerable interest has been shown in developing techniques to solve these integrals, including the following non-linear transformation techniques [1, 2, 3, 4, 5, 6] and those mentioned below.

Singh and Tanaka [7] recently studied the various non-linear transformation methods attributed to Telles [8], Sato *et al.* [9], Doblaré and Gracia [10] and Johnston [11], as applied to a simple class of weakly singular integrals. These methods all involve the use of a non-linear transformation followed by Gaussian quadrature.

Singh and Tanaka's conclusions [7] include the following:

- (1) *Non-linear transformations valid for end-point singularities perform much better than generalized transformations valid for any location of singularity. Hence, if the singular point lies in the interior of the integration element, non-linear transformations should be used in conjunction with partitioning.*
- (2) *Higher-order transformations of Sato et al. [9] represent a superset of transformations of Telles [8] and the most accurate end-point transformation of Doblaré and Gracia [10]. These transformations are also the most accurate amongst the ones considered in this study.*

It is worth noting, in relation to (1), that it is already known from theory [12] that a Gauss-Legendre quadrature rule performs better for end-point singularities than for internal singularities of the integrand function and that the order of convergence for the first case is about twice that achieved in the second case.

It is also worth noting that the higher-order transformation of Sato *et al.*, referred to in (2) above, has been shown to be equivalent to the Monomial transformation recently introduced

by one of the authors [13]. This transformation arose out of previous studies on sigmoidal transformations [11, 14], where the Monomial transformation is a limit of the $\frac{1}{m}$ th sigmoidal transformation as $m \rightarrow \infty$.

There are some advantages to using the Monomial rather than Sato's transformation: the first is that much higher transformation orders can be used, for a particular number of Gaussian points, without limits of machine precision becoming an issue [15] and the second is that the error analysis associated with sigmoidal transformations can be utilised [13, 16]. Hence the term Monomial(Sato) method is used in this paper. Details of the Monomial method are presented in Appendix 1.

Singh and Tanaka [7] also introduce a new method in the above mentioned paper, the combined Subtraction of Singularity and Non-Linear Transformation approach (SSNT) and apply it to a weakly singular integral, which arises in the solution of the two dimensional Laplace equation. They use Sato's transformation, discussed above, for the non-linear transform part of the SSNT method and also, along with Gaussian quadrature, as a basis of comparison with the SSNT method. They conclude that *'the proposed composite approach is more accurate, efficient and robust than the singularity subtraction and the non-linear transform methods'*.

Section 2 of this paper defines the weakly singular integrals being considered, while Section 3 looks at applying the SSNT method to a larger number of integrals and compares the results with the Monomial(Sato) method. It then goes on, in Section 4, to present a generalised SSNT method (GSSNT) and to study the accuracy of this new method. In Sections 5 and 6, the GSSNT, SSNT and Monomial(Sato) methods are applied to a simpler class of integrals, so that various features of the new method can be studied across the full range of singular values. Some theory associated with the GSSNT and SSNT methods is presented in Section 7 with conclusions from this study being presented in Section 8.

2 Weakly Singular Integrals

This paper is concerned with methods for solving the weakly singular integrals which arise in the solution of the two-dimensional Laplace equation. These integrals are of the form

$$I(\mathbf{x}_s, k) = \int_{\Gamma_e} \phi_k(\mathbf{x}) \ln |\mathbf{x} - \mathbf{x}_s| d\Gamma(\mathbf{x}) \quad (2.1)$$

where Γ_e is the current boundary element, \mathbf{x}_s is the collocation point and ϕ_k denotes the shape function associated with the k^{th} local node of the element. Clearly, $I(\mathbf{x}_s, k)$ is weakly singular when $\mathbf{x}_s \in \Gamma_e$.

The usual practice is to rewrite the integral in terms of the local coordinate η , $-1 \leq \eta \leq 1$, so that

$$I(\mathbf{x}_s, k) = \int_{-1}^1 \phi_k(\eta) \ln |\mathbf{x} - \mathbf{x}_s| J(\eta) d\eta \quad (2.2)$$

where J is the Jacobian of the transformation of Γ_e onto $[-1, 1]$.

3 SSNT Approach

Recently, Singh and Tanaka [7] introduced a new combined approach to the solution of weakly singular integrals, the so-called SSNT approach (Singularity Subtraction followed by a Non-linear Transformation).

The idea behind the SSNT approach is to re-write the integral as the sum of a singular integral, which can be evaluated analytically, and a regular integral, which could be evaluated using Gaussian quadrature but is, in fact, evaluated by first using a non-linear transformation to obtain a smoother integrand, and then by Gaussian quadrature. See [7] for a full explanation of the method.

When the SSNT approach is applied to the integrals given in equation (2.2), Singh and Tanaka suggest two possible choices for the function $f_s(\eta, \eta_s)$ to be subtracted from the inte-

grand:

(1) The (LOGA) approach, where $f_s(\eta, \eta_s) = J(\eta_s) \ln |\mathbf{x} - \mathbf{x}_s|$, giving

$$I(\mathbf{x}_s, k) = \int_{-1}^1 J(\eta_s) \ln |\mathbf{x} - \mathbf{x}_s| d\eta + \int_{-1}^1 [\phi_k(\eta)J(\eta) - J(\eta_s)] \ln |\mathbf{x} - \mathbf{x}_s| d\eta \quad (3.1)$$

and

(2) the (LOGB) approach, where $f_s(\eta, \eta_s) = J(\eta_s) \ln |\eta - \eta_s|$, giving

$$I(\mathbf{x}_s, k) = \int_{-1}^1 J(\eta_s) \ln |\eta - \eta_s| d\eta + \int_{-1}^1 (\phi_k(\eta)J(\eta) \ln |\mathbf{x} - \mathbf{x}_s| - J(\eta_s) \ln |\eta - \eta_s|) d\eta \quad (3.2)$$

In each case the first integral is evaluated analytically and the second integral is evaluated by applying a non-linear transformation, that of Sato *et al.* [9] of order 3 or 4, followed by Gaussian quadrature.

3.1 Percentage Relative Error

The various methods considered here will be compared in terms of the Percentage Relative Error, defined by

$$\text{Percentage Relative Error} = \frac{|I_{\text{approximate}} - I_{\text{exact}}|}{I_{\text{exact}}} \times 100 \quad (3.3)$$

where I_{exact} is found using Mathematica [17]. Double precision arithmetic has been used in numerical calculations.

3.2 SSNT Results

In their paper [7], Singh and Tanaka consider a curved quadratic element with nodes $\mathbf{x}_1=(1,1)$, $\mathbf{x}_2=(2.5,3)$ and $\mathbf{x}_3=(2,5)$, in which \mathbf{x}_2 is the middle node and this will be used here.

Table 1 includes the case considered by Singh and Tanaka, that is $I(\mathbf{x}_3, 3)$, but a wider range of integrals is included; that is, $I(\mathbf{x}_1, 1)$, $I(\mathbf{x}_2, 2)$ and $I(\mathbf{x}_3, 3)$, as well as the quasi-singular [7] integrals $I(\mathbf{x}_1, 2)$, $I(\mathbf{x}_1, 3)$, $I(\mathbf{x}_2, 1)$, $I(\mathbf{x}_2, 3)$, $I(\mathbf{x}_3, 1)$ and $I(\mathbf{x}_3, 2)$. Note that it is worth applying

a transformation to quasi-singular integrals as, although they are bounded at the singular point of the logarithm function, the integral exhibits rapid change near that point. It is well known that clustering integration points in a region where the integrand changes rapidly improves the accuracy of the numerical approximation to the value of the integral.

Comparisons of Percentage Relative Errors between SSNT(LOGA) and SSNT(LOGB) with orders 3 and 4 and the Monomial(Sato) method as well as Gaussian quadrature are presented for various numbers of Gaussian points, NG.

Perusal of Table 1 indicates:

- (1) SSNT(LOG A) and SSNT(LOG B) yield very similar results.
- (2) SSNT of order 3 is ‘best’ (gives the lowest Percentage Relative Errors) for the weakly singular integrals $I(\mathbf{x}_k, k)$ where $k=1, 2, 3$.
- (3) The Monomial(Sato) method is almost always ‘best’ for the rest of the integrals (that is, the quasi-singular integrals) except sometimes when NG=6 when either SSNT or Gaussian quadrature is best.

4 A Generalised SSNT Approach

Since the SSNT method produces superior results to the Monomial(Sato) method for only the three nodes where $\phi_k(\eta_k) = 1$, that is $I(\mathbf{x}_1, 1)$, $I(\mathbf{x}_2, 2)$ and $I(\mathbf{x}_3, 3)$, it would be useful if a modification to this method could be found which would take account of the behaviour of $\phi_k(\eta)$ and therefore be applicable over the whole range of η_s values, $\eta_s \in [-1, 1]$.

Consideration of equation (3.1) for $I(\mathbf{x}_k, k)$ gives

$$\begin{aligned}
I(\mathbf{x}_k, k) &= \int_{-1}^1 1 * J(\eta_k) \ln |\mathbf{x} - \mathbf{x}_k| d\eta + \int_{-1}^1 (\phi_k(\eta) * J(\eta) - 1 * J(\eta_k)) \ln |\mathbf{x} - \mathbf{x}_k| d\eta \\
&= \int_{-1}^1 \phi_k(\eta_k) * J(\eta_k) \ln |\mathbf{x} - \mathbf{x}_k| d\eta \\
&\quad + \int_{-1}^1 (\phi_k(\eta) * J(\eta) - \phi_k(\eta_k) * J(\eta_k)) \ln |\mathbf{x} - \mathbf{x}_k| d\eta
\end{aligned} \tag{4.1}$$

and this leads to the generalised SSNT(LOGA) method, GSSNT(LOGA):

$$\begin{aligned}
I(\mathbf{x}_k, k) &= \int_{-1}^1 \phi_k(\eta_s) * J(\eta_s) \ln |\mathbf{x} - \mathbf{x}_s| d\eta \\
&\quad + \int_{-1}^1 (\phi_k(\eta) * J(\eta) - \phi_k(\eta_s) * J(\eta_s)) \ln |\mathbf{x} - \mathbf{x}_s| d\eta \\
&= I_{SA}(\mathbf{x}_s, k) + I_{RA}(\mathbf{x}_s, k)
\end{aligned} \tag{4.2}$$

where I_{SA} is the singular integral and I_{RA} is the regular integral mentioned in Section 3.

Expressions for $\ln |\mathbf{x} - \mathbf{x}_s|$ and $J(\eta)$ are given in Section 5 for the curved element used in Section 3. Using these a value for $\int_{-1}^1 \ln |\mathbf{x} - \mathbf{x}_s| d\eta$ is found using Mathematica [17] and hence I_{SA} is evaluated. The regular integral I_{RA} is calculated using the Monomial(Sato) method outlined in the Appendix.

A similar approach is followed for the generalised SSNT(LOGB) method, GSSNT(LOGB); using equation (3.2),

$$\begin{aligned}
I(\mathbf{x}_k, k) &= \int_{-1}^1 \phi_k(\eta_s) * J(\eta_s) \ln |\eta - \eta_s| d\eta \\
&\quad + \int_{-1}^1 (\phi_k(\eta) * J(\eta) \ln |\mathbf{x} - \mathbf{x}_s| - \phi_k(\eta_s) * J(\eta_s) \ln |\eta - \eta_s|) d\eta \\
&= I_{SB}(\mathbf{x}_s, k) + I_{RB}(\mathbf{x}_s, k)
\end{aligned} \tag{4.3}$$

where again I_{SB} is the singular integral and I_{RB} is the regular integral mentioned in Section 3.

Note that here $\int_{-1}^1 \ln |\eta - \eta_s| d\eta$ and hence I_{SB} can be found analytically. Clearly, this is a major advantage of the (LOGB) technique over the (LOGA) technique.

It can be seen from equations (4.1) and (4.2), that for the integrals $I(\mathbf{x}_k, k)$ where $k=1, 2, 3$, $\phi_k(\eta_s) = \phi_k(\eta_k) = 1$ and GSSNT is equivalent to SSNT. In the case of the quasi-singular

integrals $I(\mathbf{x}_s, k)$ ($s \neq k$ where $s, k = 1, 2, 3$), $\phi_k(\eta_s) = 0$ and GSSNT simply becomes the Monomial(Sato) method. This neatly accounts for the results of Section 3, which found that SSNT is superior for the weakly singular integrals and that Monomial(Sato) is almost always superior for the quasi-singular integrals.

4.1 GSSNT Results

Table 2 presents results taken from Table 1 and compares them with results for GSSNT for the (LOGB) case of order 3. Only order 3 and (LOGB) results are given since the other results are already available in Table 1 and also because Table 1 shows that (LOGA) and (LOGB) results are very similar. The Table 2 results show that GSSNT is superior, or equal, to both the Monomial(Sato) method and SSNT, in all but two cases where NG=6. This was, of course, to be expected, as GSSNT defaults to either the SSNT method or the Monomial(Sato) method at the node points.

So far all the cases considered have involved a singularity at one of the node points. In the case where $(\eta_s \neq -1, 0, 1)$ it is obviously more difficult to calculate the ‘exact’ value of the integral and also I_{SA} and so only two cases are presented here. As mentioned previously, I_{SB} , however, is only a multiple of $\int_{-1}^1 \ln |\eta - \eta_s| d\eta$ and can therefore be found easily (analytically).

Table 3 gives results for two cases, $\eta_s=0.5$ and $\eta_s=-0.1$, where η_s is not a node point and therefore $\phi(\eta_s) \neq 0$ or 1. The results indicate that GSSNT is superior or equal to both SSNT and the Monomial(Sato) method, in all but one case where NG=6. Also, note that order 3 results are better than order 4 results.

5 A Simpler Case

In this section, a study is made comparing the GSSNT, SSNT and the Monomial(Sato) methods applied to a simpler version of the integrals considered in the previous section,

$$I_0(\eta_s, k) = \int_{-1}^1 \phi_k(\eta) \ln |\eta - \eta_s| d\eta \quad (5.1)$$

where the Jacobian $J(\eta) \equiv 1$ and the kernel $\ln |\eta - \eta_s|$ is used.

The purpose of this simplification is so that the complete range of values of $\eta_s \in [-1, 1]$ can be considered, rather than primarily the node points as in Sections 3 and 4. The aim here is to first learn about the range across which GSSNT is superior to the other two methods. A later task will be to study the non-linear transformation order and its effect on the relative errors of the evaluated integrals.

Clearly, the (LOGA) and (LOGB) techniques are identical for $I_0(\eta_s, k)$ and so Figures 1 and 2 show the Percentage Relative Error for each of the three methods GSSNT, SSNT and Monomial(Sato), all of order 4 and NG=10 for $I_0(\eta_s, k)$, where η_s ranges from -1 to 1, for $k=1$ and $k=2$ respectively. The graph for $k=3$ is simply the reflection in η_s about 0 of the graph for $k=1$, since $\phi_1(-\eta) = \frac{1}{2}(-\eta)(1 - (-\eta)) = -\frac{1}{2}(\eta)(1 + \eta) = -\phi_3(\eta)$ and so this is omitted.

Figures 1 and 2 show that GSSNT produces relative errors smaller than or equal to both other methods for *all* η_s for 10 Gaussian points. Similar graphs show that for NG=6, GSSNT is superior to or equal to SSNT for all η_s and that GSSNT is generally superior to the Monomial(Sato) method except near 1 for ϕ_1 and near ± 1 for ϕ_2 .

Note that in Figures 1 and 2 for ϕ_1 , SSNT=GSSNT at $\eta_s=-1$ and for ϕ_2 , SSNT=GSSNT at $\eta_s=0$. This is because at these node points $\phi_1(-1) = \phi_2(0) = \phi_3(1) = 1$ and as previously discussed in Section 4, this means that the GSSNT and SSNT methods are identical at that point. It can also be seen that results from Monomial(Sato) and GSSNT are equal at $\eta_s=0$ and 1 for ϕ_1 and at $\eta_s=-1$ and 1 for ϕ_2 . This is due to the fact that for these values of η_s

$\phi_k(\eta_s) = 0$, which causes the GSSNT and Monomial(Sato) methods to be identical as pointed out in Section 4.

6 ‘Optimal’ Transformation Orders

The final question to be considered relates to the order which should be chosen for the Monomial(Sato) transformation in the GSSNT method. Since the method defaults to either the Monomial(Sato) method or the SSNT method at the node points, but not elsewhere, it is worth beginning with previously drawn conclusions relating to the other methods. Singh and Tanaka [7] suggest that for the Monomial(Sato) method a conservative choice would be order 4 or 5 for double precision calculations and that the use of order 3 would be sufficient for the SSNT method.

It, therefore, seems worthwhile to consider the GSSNT method with transformation orders 3, 4 and 5 as applied to $I_0(\eta_s, k)$. Both Figures 3 and 4 do this for NG=6 and ϕ_1 , and NG=10 and ϕ_2 respectively, with order 6 being included in the 10 Gaussian point case as well. Figure 3 shows that for six Gaussian points, order 3 produces superior results to order 4 (and order 5) for all values of η_s , and the same was found to be true for ϕ_2 . The situation is somewhat more complex for 10 Gaussian points. Figure 4 shows that order 5 produces superior results to order 4 (and 3) but that results for order 6 are not always superior to those of order 5. Similar remarks apply to the NG=10 and ϕ_1 case.

Studies for higher numbers of Gaussian points show that even order 3 and 4 transformations give Truncation Errors ($|I_{exact} - I_{approximate}|$) near the limits of machine precision.

Based on these observations, as well as Tables 2 and 3, it would seem that, in general, a transformation order of 3 is quite sufficient for the GSSNT method to produce very accurate evaluations of weakly singular integrals.

7 Theory

When Singh and Tanaka [7] introduced the SSNT method, they pointed out that the function $f_r(\eta, \eta_s)$, created by the subtraction of singularity, is now regular, that is, $\lim_{\eta \rightarrow \eta_s} f_r(\eta, \eta_s) = 0$ (or C where C is a bounded constant). Thus $I_r(\mathbf{x}_s, k) = \int_{-1}^1 f_r(\eta, \eta_s) d\eta$ can be evaluated using Gaussian quadrature; but it is, in fact, integrated using a Monomial(Sato) transformation to obtain a smoother integrand followed by Gaussian quadrature.

Now in the case of the GSSNT(LOGA) method, using equation (4.2) gives

$$f_r(\eta, \eta_s) = [\phi_k(\eta)J(\eta) - \phi_k(\eta_s)J(\eta_s)] \ln |\mathbf{x} - \mathbf{x}_s| \quad (7.1)$$

and for the case of the curved quadratic element used in Section 3.2

$$\ln |\mathbf{x} - \mathbf{x}_s| = \ln |\eta - \eta_s| + \frac{1}{2} \ln \left[(\eta + \eta_s)(\eta + \eta_s - 1) + \frac{17}{4} \right] \quad (7.2)$$

and

$$J(\eta) = \sqrt{4\eta^2 - 2\eta + \frac{17}{4}} = 2 \left[1 + \left(\eta - \frac{1}{4} \right)^2 \right]^{\frac{1}{2}} \quad (7.3)$$

Then it can be shown that expanding $J(\eta)$ using a Binomial series gives

$$\phi_k(\eta)J(\eta) - \phi_k(\eta_s)J(\eta_s) = (\eta - \eta_s)g_k(\eta, \eta_s) \quad (7.4)$$

where $\lim_{\eta \rightarrow \eta_s} g_k(\eta, \eta_s) = C$ for C a constant and $k = 1, 2, 3$. Hence f_r is regular since $\lim_{\eta \rightarrow \eta_s} f_r(\eta, \eta_s) = 0$.

For the GSSNT(LOGB) method

$$f_r(\eta, \eta_s) = \phi_k(\eta)J(\eta) \ln |\mathbf{x} - \mathbf{x}_s| - \phi_k(\eta_s)J(\eta_s) \ln |\eta - \eta_s| \quad (7.5)$$

and using a similar approach to the above gives $\lim_{\eta \rightarrow \eta_s} f_r(\eta, \eta_s) = c$ for c a constant, and thus f_r is regular in this case also.

The fact that f_r is regular for both the GSSNT(LOGA) and GSSNT(LOGB) methods can also be shown for a general curved element.

This is not the case, however, for the SSNT method in general, although for the three cases where SSNT works well, that is at the three nodes where $\phi_k(\eta_k) = 1$, it can be shown similarly to the above, that f_r is regular.

8 Conclusions

This paper has introduced a generalisation of the SSNT (Subtraction of Singularity followed by a Non-Linear Transformation) method, recently introduced by Singh and Tanaka [7], to improve the accuracy of evaluating weakly singular boundary integrals. Both methods introduce a function, $f_s(\eta, \eta_s)$, of the form, constant multiplied by a logarithm function, for the subtraction of singularity, but the new generalised SSNT method (GSSNT) takes into account the behaviour of the basis functions, whereas the SSNT method does not. The regular integral that remains after the subtraction of singularity, is then evaluated, for both methods, by splitting the interval at the singularity and then applying a Monomial(Sato) transformation followed by Gaussian quadrature.

The GSSNT method has been implemented and numerically compared with both the SSNT and the Monomial(Sato) methods for the weakly singular integrals which arise in the boundary element solution to the two dimensional Laplace equation. In almost every case considered, the new technique is either equivalent or superior to both the SSNT and the Monomial(Sato) methods, bearing in mind that prior to Singh and Tanaka's introduction of the SSNT method [7], the Monomial(Sato) method was regarded as the preferred technique for these integrals (see (2) in the Introduction).

It is worth noting that in both the GSSNT and SSNT methods, there are two possible choices for the function $f_s(\eta, \eta_s)$: the so-called (LOGA) version, where $f_s(\eta, \eta_s) = c * \ln |\mathbf{x}_s - \mathbf{x}|$ and the (LOGB) version which involves $f_s(\eta, \eta_s) = c * \ln |\eta - \eta_s|$, where c is a constant. Since

it is possible to analytically integrate $\int_{-1}^1 \ln|\eta - \eta_s| d\eta$ and the results for the two techniques are almost identical in the cases considered, (LOGB) is clearly the more easily implemented technique of the two.

The question of the ‘optimal’ order of the transformation to be used for the Monomial(Sato) transformation part of the GSSNT method was also considered. It was found that a transformation order of three is sufficient to produce very accurate results for the evaluation of weakly singular boundary integrals.

Given that the GSSNT method is so successful in evaluating one dimensional singular integrals, it might be worth extending it to two dimensions to determine its applicability and accuracy there. Possible applications would be to two different types of weakly singular boundary integrals: those which arise when considering the three dimensional Laplace equation and also those involved in the solution of Poisson’s equation in two dimensions.

In summary then, the GSSNT method presented above almost always yields more accurate values, for the weakly singular integrals considered, than both the SSNT and the non-linear transformation methods, with the (LOGB) version of the technique being preferred because of its ease of implementation.

Appendix

Monomial Method

To apply the Monomial method [13] to a one dimensional integral with an arbitrary singularity $\eta_s \in (-1, 1)$, firstly split the interval at the singularity, then change the variable of integration so that both integrals are evaluated over $[0, 1]$, with the singularity mapping to 0 in both cases, and then apply the Monomial transformation to each of the resulting integrals.

For example, in the case of the weakly singular integrals given in equation (2.2),

$$I(\mathbf{x}_s, k) = \int_{-1}^{\eta_s} \phi_k(\eta) \ln |\mathbf{x} - \mathbf{x}_s| J(\eta) d\eta + \int_{\eta_s}^1 \phi_k(\eta) \ln |\mathbf{x} - \mathbf{x}_s| J(\eta) d\eta$$

Now change the variable of integration to evaluate both integrals over the range $[0,1]$, ensuring that the point η_s maps to 0 in each case. That is, for the first integral in the equation above, apply $\eta = \eta_s - (1 + \eta_s)t$ and for the second apply, $\eta = \eta_s + (1 - \eta_s)t$, giving

$$\begin{aligned} I(\mathbf{x}_s, k) = & (1 + \eta_s) \int_0^1 \phi_k(\eta_s - (1 + \eta_s)t) \ln |\mathbf{x}(\eta_s - (1 + \eta_s)t) - \mathbf{x}_s| J(\eta_s - (1 + \eta_s)t) dt \\ & + (1 - \eta_s) \int_0^1 \phi_k(\eta_s + (1 - \eta_s)t) \ln |\mathbf{x}(\eta_s + (1 - \eta_s)t) - \mathbf{x}_s| J(\eta_s + (1 - \eta_s)t) dt \end{aligned}$$

Next a Monomial transformation of order n is applied, where $t = \gamma(w; n) = w^n$, giving

$$\begin{aligned} I(\mathbf{x}_s, k) = & (1 + \eta_s) \int_0^1 \phi_k(\eta_s - (1 + \eta_s)w^n) \ln |\mathbf{x}(\eta_s - (1 + \eta_s)w^n) - \mathbf{x}_s| J(\eta_s - (1 + \eta_s)w^n) n w^{n-1} dw \\ & + (1 - \eta_s) \int_0^1 \phi_k(\eta_s + (1 - \eta_s)w^n) \ln |\mathbf{x}(\eta_s + (1 - \eta_s)w^n) - \mathbf{x}_s| J(\eta_s + (1 - \eta_s)w^n) n w^{n-1} dw \end{aligned}$$

Finally, apply Gaussian quadrature to each of the above integrals.

References

- [1] M. Cerrolaza and E. Alarcón. A bi-cubic transformation for the numerical evaluation of the Cauchy principal value integrals in boundary methods. *International Journal for Numerical Methods in Engineering*, 28:987–999, 1989.
- [2] J. Sanz-Serna, M. Doblaré, and E. Alarcón. Remarks on methods for the computation of boundary-element integrals by co-ordinate transformation. *Communications in Numerical Methods in Engineering.*, 6:121–123, 1990.
- [3] James H. Kane. *Boundary Element Analysis in Engineering Continuum Mechanics*. Prentice Hall, 1994.
- [4] G. Monegato and I. H. Sloan. Numerical solution of the generalised airfoil equation for an airfoil with a flap. *SIAM Journal of Numerical Analysis*, 34(6):2288–2305, 1997.

- [5] L. Scuderi. A collocation method for the generalised airfoil equation for an airfoil with a flap. *SIAM Journal of Numerical Analysis*, 35(5):1725–1739, 1998.
- [6] Peter R. Johnston and David Elliott. A generalisation of Telles’ method for evaluating weakly singular boundary element integrals. *Journal of Computational and Applied Mathematics*, 131:223–241, 2001.
- [7] K.M. Singh and M. Tanaka. On non-linear transformations for accurate numerical evaluation of weakly singular boundary integrals. *International Journal for Numerical Methods in Engineering*, 50:2007–2030, 2001.
- [8] J. C. F. Telles. A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals. *International Journal for Numerical Methods in Engineering*, 24:959–973, 1987.
- [9] M. Sato, S. Yoshioka, and K. Tsukui. Accurate numerical integration of singular kernels in the two-dimensional boundary element method. In C. A. Brebbia, editor, *Boundary Elements X, Vol 1*, pages 279–298. Springer-Verlag, 1988.
- [10] M. Doblaré and L. Gracia. On non-linear transformations for the integration of weakly-singular and Cauchy principal value integrals. *International Journal for Numerical Methods in Engineering*, 40:3325–3358, 1997.
- [11] Peter R. Johnston. Application of sigmoidal transformations to weakly singular and near singular boundary element integrals. *International Journal for Numerical Methods in Engineering*, 45(10):1333–1348, August 1999.
- [12] G. Monegato and L. Scuderi. Numerical integration of functions with boundary singularities. *Journal of Computational and Applied Mathematics*, 112:201–214, 1999.

- [13] Peter R. Johnston and David Elliott. Transformations for evaluating singular boundary element integrals. *Journal of Computational and Applied Mathematics*, 146(2):231–251, 2002.
- [14] Peter R. Johnston. Semi-sigmoidal transformations for evaluating weakly singular boundary element integrals. *International Journal for Numerical Methods in Engineering*, 47(10):1709–1730, 2000.
- [15] Peter R. Johnston and Barbara M. Johnston. A simple device to improve the accuracy of evaluating weakly singular boundary element integrals. *Communications in Numerical Methods in Engineering*, 18(3):189–194, 2002.
- [16] Peter R. Johnston and David Elliott. Error estimation of quadrature rules for evaluating singular integrals in boundary element problems. *International Journal for Numerical Methods in Engineering*, 48(7):949–962, 2000.
- [17] Stephen Wolfram. *The Mathematica Book*. Wolfram Media/Cambridge University Press, 3rd edition, 1996.

Captions

Table 1: Percentage Relative Errors for various methods used to evaluate $I(\mathbf{x}_s, k) = \int_{-1}^1 \phi_k(\eta) \ln |\mathbf{x} - \mathbf{x}_s| J(\eta) d\eta$ at the node points for transformation orders 3 and 4, for all basis functions and various numbers of Gaussian points.

Table 2: Percentage Relative Errors for the GSSNT, SSNT and Monomial(Sato) methods, all of order 3, used to evaluate $I(\mathbf{x}_s, k) = \int_{-1}^1 \phi_k(\eta) \ln |\mathbf{x} - \mathbf{x}_s| J(\eta) d\eta$ at the node points, for all basis functions and various numbers of Gaussian points.

Table 3: Percentage Relative Errors for evaluating $I(\mathbf{x}_s, k) = \int_{-1}^1 \phi_k(\eta) \ln |\mathbf{x} - \mathbf{x}_s| J(\eta) d\eta$ using the GSSNT, SSNT and Monomial(Sato) methods, for various numbers of Gaussian points for $\eta_s = 0.5$ and ϕ_1 , and $\eta_s = -0.1$ and ϕ_2 respectively.

Figure 1: Graph of Percentage Relative Errors for the GSSNT, SSNT and Monomial(Sato) methods evaluating $I_0(\eta_s, 1) = \int_{-1}^1 \phi_1(\eta) \ln |\eta - \eta_s| d\eta$ for η_s varying between -1 and 1, 10 Gaussian points and transformation order 4.

Figure 2: Graph of Percentage Relative Errors for the GSSNT, SSNT and Monomial(Sato) methods evaluating $I_0(\eta_s, 2) = \int_{-1}^1 \phi_2(\eta) \ln |\eta - \eta_s| d\eta$ for η_s varying between -1 and 1, 10 Gaussian points and transformation order 4.

Figure 3: Graph of Percentage Relative Errors for the GSSNT method evaluating $I_0(\eta_s, 1) = \int_{-1}^1 \phi_1(\eta) \ln |\eta - \eta_s| d\eta$ for η_s varying between -1 and 1, 6 Gaussian points and transformation orders 3, 4 and 5.

Figure 4: Graph of Percentage Relative Errors for the GSSNT method evaluating $I_0(\eta_s, 2) = \int_{-1}^1 \phi_2(\eta) \ln |\eta - \eta_s| d\eta$ for η_s varying between -1 and 1, 10 Gaussian points and transformation orders 3, 4, 5 and 6.

η_s	ϕ_i	NG	Percentage Relative Errors						
			SSNT (LOGA) order 3	SSNT (LOGA) order 4	SSNT (LOGB) order 3	SSNT (LOGB) order 4	Monomial (Sato) order 3	Monomial (Sato) order 4	Gaussian Quad.
-1	ϕ_1	6	9.2×10^{-3}	9.3×10^{-1}	1.3×10^{-2}	9.4×10^{-1}	4.1×10^{-2}	9.4×10^{-1}	1.0×10^1
		10	1.8×10^{-4}	3.2×10^{-3}	1.8×10^{-4}	3.2×10^{-3}	1.7×10^{-3}	3.3×10^{-3}	3.9×10^0
		20	9.7×10^{-10}	4.2×10^{-8}	9.7×10^{-10}	4.2×10^{-8}	2.7×10^{-5}	3.6×10^{-7}	1.0×10^0
	ϕ_2	6	1.1×10^{-1}	8.1×10^{-1}	1.1×10^{-1}	8.0×10^{-1}	9.8×10^{-2}	8.0×10^{-1}	1.8×10^{-1}
		10	1.2×10^{-3}	4.2×10^{-3}	1.2×10^{-3}	4.2×10^{-3}	5.6×10^{-4}	4.2×10^{-3}	2.5×10^{-2}
		20	1.2×10^{-5}	1.5×10^{-7}	1.2×10^{-5}	1.5×10^{-7}	2.8×10^{-9}	1.6×10^{-8}	1.7×10^{-3}
	ϕ_3	6	8.7×10^{-2}	4.7×10^{-1}	8.9×10^{-2}	4.7×10^{-1}	1.1×10^{-1}	4.8×10^{-1}	8.7×10^{-2}
		10	6.7×10^{-4}	2.1×10^{-3}	6.7×10^{-4}	2.1×10^{-3}	5.8×10^{-4}	2.2×10^{-3}	1.2×10^{-2}
		20	2.2×10^{-5}	2.9×10^{-7}	2.2×10^{-5}	2.9×10^{-7}	2.4×10^{-9}	3.2×10^{-8}	7.8×10^{-4}
0	ϕ_1	6	3.1×10^{-2}	6.9×10^{-2}	3.1×10^{-2}	6.9×10^{-2}	4.5×10^{-4}	7.3×10^{-2}	2.2×10^{-3}
		10	1.7×10^{-3}	7.4×10^{-5}	1.7×10^{-3}	7.4×10^{-5}	2.5×10^{-7}	3.1×10^{-6}	1.2×10^{-4}
		20	3.0×10^{-5}	3.5×10^{-7}	3.0×10^{-5}	3.5×10^{-7}	4.0×10^{-13}	4.2×10^{-13}	2.6×10^{-6}
	ϕ_2	6	1.3×10^{-3}	5.3×10^{-2}	1.3×10^{-3}	5.3×10^{-2}	1.0×10^{-2}	5.2×10^{-2}	4.0×10^0
		10	4.4×10^{-8}	2.8×10^{-7}	4.4×10^{-8}	2.4×10^{-7}	6.1×10^{-4}	2.8×10^{-5}	1.5×10^0
		20	1.9×10^{-13}	4.3×10^{-14}	2.4×10^{-13}	1.4×10^{-14}	1.1×10^{-5}	1.2×10^{-7}	4.0×10^{-1}
	ϕ_3	6	6.2×10^{-2}	1.4×10^{-1}	6.2×10^{-2}	1.4×10^{-1}	3.4×10^{-3}	1.5×10^{-1}	2.8×10^{-3}
		10	3.5×10^{-3}	1.7×10^{-4}	3.5×10^{-3}	1.7×10^{-4}	5.3×10^{-7}	8.4×10^{-6}	1.5×10^{-4}
		20	6.2×10^{-5}	7.2×10^{-7}	6.2×10^{-5}	7.2×10^{-7}	3.9×10^{-13}	1.2×10^{-13}	2.6×10^{-6}
1	ϕ_1	6	5.8×10^{-3}	3.5×10^{-1}	5.7×10^{-3}	3.3×10^{-1}	2.1×10^{-2}	3.3×10^{-1}	5.5×10^{-2}
		10	9.6×10^{-4}	9.5×10^{-4}	9.5×10^{-4}	9.7×10^{-4}	1.6×10^{-4}	9.3×10^{-4}	7.5×10^{-3}
		20	1.4×10^{-5}	1.6×10^{-7}	1.4×10^{-5}	1.6×10^{-7}	1.4×10^{-10}	2.0×10^{-9}	5.0×10^{-4}
	ϕ_2	6	8.1×10^{-2}	5.6×10^{-1}	8.1×10^{-2}	5.8×10^{-1}	6.9×10^{-2}	5.7×10^{-1}	1.6×10^{-1}
		10	3.5×10^{-4}	1.8×10^{-3}	3.5×10^{-4}	1.8×10^{-3}	2.6×10^{-4}	1.8×10^{-3}	2.3×10^{-2}
		20	1.1×10^{-5}	1.3×10^{-7}	1.1×10^{-5}	1.3×10^{-7}	7.0×10^{-10}	7.1×10^{-9}	1.5×10^{-3}
	ϕ_3	6	6.5×10^{-2}	7.2×10^{-1}	6.5×10^{-2}	6.8×10^{-1}	8.5×10^{-2}	6.9×10^{-1}	7.6×10^0
		10	1.2×10^{-5}	5.2×10^{-4}	1.1×10^{-5}	5.4×10^{-4}	1.1×10^{-3}	4.9×10^{-4}	2.8×10^0
		20	5.5×10^{-10}	1.5×10^{-8}	5.5×10^{-10}	1.5×10^{-8}	2.0×10^{-5}	2.4×10^{-7}	7.3×10^{-1}

Table 1:

η_s	ϕ_i	NG	Percentage Relative Errors		
			GSSNT(LOG B)	SSNT(LOG B)	Monomial(Sato)
-1	ϕ_1	6	1.3×10^{-2}	1.3×10^{-2}	4.1×10^{-2}
		10	1.8×10^{-4}	1.8×10^{-4}	1.7×10^{-3}
		20	9.7×10^{-10}	9.7×10^{-10}	2.7×10^{-5}
	ϕ_2	6	9.8×10^{-2}	1.1×10^{-1}	9.8×10^{-2}
		10	5.6×10^{-4}	1.2×10^{-3}	5.6×10^{-4}
		20	2.8×10^{-9}	1.2×10^{-5}	2.8×10^{-9}
	ϕ_3	6	1.1×10^{-1}	8.9×10^{-2}	1.1×10^{-1}
		10	5.8×10^{-4}	6.7×10^{-4}	5.8×10^{-4}
		20	2.4×10^{-9}	2.2×10^{-5}	2.4×10^{-9}
0	ϕ_1	6	4.5×10^{-4}	3.1×10^{-2}	4.5×10^{-4}
		10	2.5×10^{-7}	1.7×10^{-3}	2.5×10^{-7}
		20	4.0×10^{-13}	3.0×10^{-5}	4.0×10^{-13}
	ϕ_2	6	1.3×10^{-3}	1.3×10^{-3}	1.0×10^{-2}
		10	4.4×10^{-8}	4.4×10^{-8}	6.1×10^{-4}
		20	2.4×10^{-13}	2.4×10^{-13}	1.1×10^{-5}
	ϕ_3	6	3.4×10^{-3}	6.2×10^{-2}	3.4×10^{-3}
		10	5.3×10^{-7}	3.5×10^{-3}	5.3×10^{-7}
		20	3.9×10^{-13}	6.2×10^{-5}	3.9×10^{-13}
1	ϕ_1	6	2.1×10^{-2}	5.7×10^{-3}	2.1×10^{-2}
		10	1.6×10^{-4}	9.5×10^{-4}	1.6×10^{-4}
		20	1.4×10^{-10}	1.4×10^{-5}	1.4×10^{-10}
	ϕ_2	6	6.9×10^{-2}	8.1×10^{-2}	6.9×10^{-2}
		10	2.6×10^{-4}	3.5×10^{-4}	2.6×10^{-4}
		20	7.0×10^{-10}	1.1×10^{-5}	7.0×10^{-10}
	ϕ_3	6	6.5×10^{-2}	6.5×10^{-2}	8.5×10^{-2}
		10	1.1×10^{-5}	1.1×10^{-5}	1.1×10^{-3}
		20	5.5×10^{-10}	5.5×10^{-10}	2.0×10^{-5}

Table 2:

Methods	Percentage Relative Errors					
	$\int_{-1}^1 \phi_1(\eta) \ln \eta - 0.5 J(\eta) d\eta$			$\int_{-1}^1 \phi_2(\eta) \ln \eta + 0.1 J(\eta) d\eta$		
	NG=6	NG=10	NG=20	NG=6	NG=10	NG=20
GSSNT(LOGA) order 3	3.1×10^{-3}	2.1×10^{-6}	5.1×10^{-14}	2.1×10^{-3}	3.5×10^{-7}	1.3×10^{-13}
GSSNT(LOGB) order 3	3.1×10^{-3}	2.1×10^{-6}	3.4×10^{-14}	2.1×10^{-3}	3.5×10^{-7}	2.7×10^{-13}
SSNT(LOGA) order 3	1.1×10^{-2}	8.1×10^{-4}	1.4×10^{-5}	2.2×10^{-3}	6.6×10^{-6}	1.1×10^{-7}
SSNT(LOGB) order 3	1.2×10^{-2}	8.1×10^{-4}	1.4×10^{-5}	2.2×10^{-3}	6.6×10^{-6}	1.1×10^{-7}
Monomial(Sato) order 3	1.5×10^{-3}	9.2×10^{-5}	1.6×10^{-6}	9.5×10^{-3}	6.2×10^{-4}	1.1×10^{-5}
GSSNT(LOGA) order 4	8.6×10^{-2}	1.9×10^{-5}	7.2×10^{-12}	6.3×10^{-2}	1.1×10^{-6}	5.2×10^{-13}
GSSNT(LOGB) order 4	8.6×10^{-2}	1.9×10^{-5}	7.1×10^{-12}	6.4×10^{-2}	1.2×10^{-6}	8.8×10^{-13}
SSNT(LOGA) order 4	8.4×10^{-2}	5.5×10^{-5}	1.6×10^{-7}	6.3×10^{-2}	1.4×10^{-6}	1.3×10^{-9}
SSNT(LOGB) order 4	8.4×10^{-2}	5.5×10^{-5}	1.6×10^{-7}	6.3×10^{-2}	1.5×10^{-6}	1.3×10^{-9}
Monomial(Sato) order 4	8.6×10^{-2}	2.3×10^{-5}	1.8×10^{-8}	6.2×10^{-2}	2.7×10^{-5}	1.3×10^{-7}

Table 3:

Figure 1:

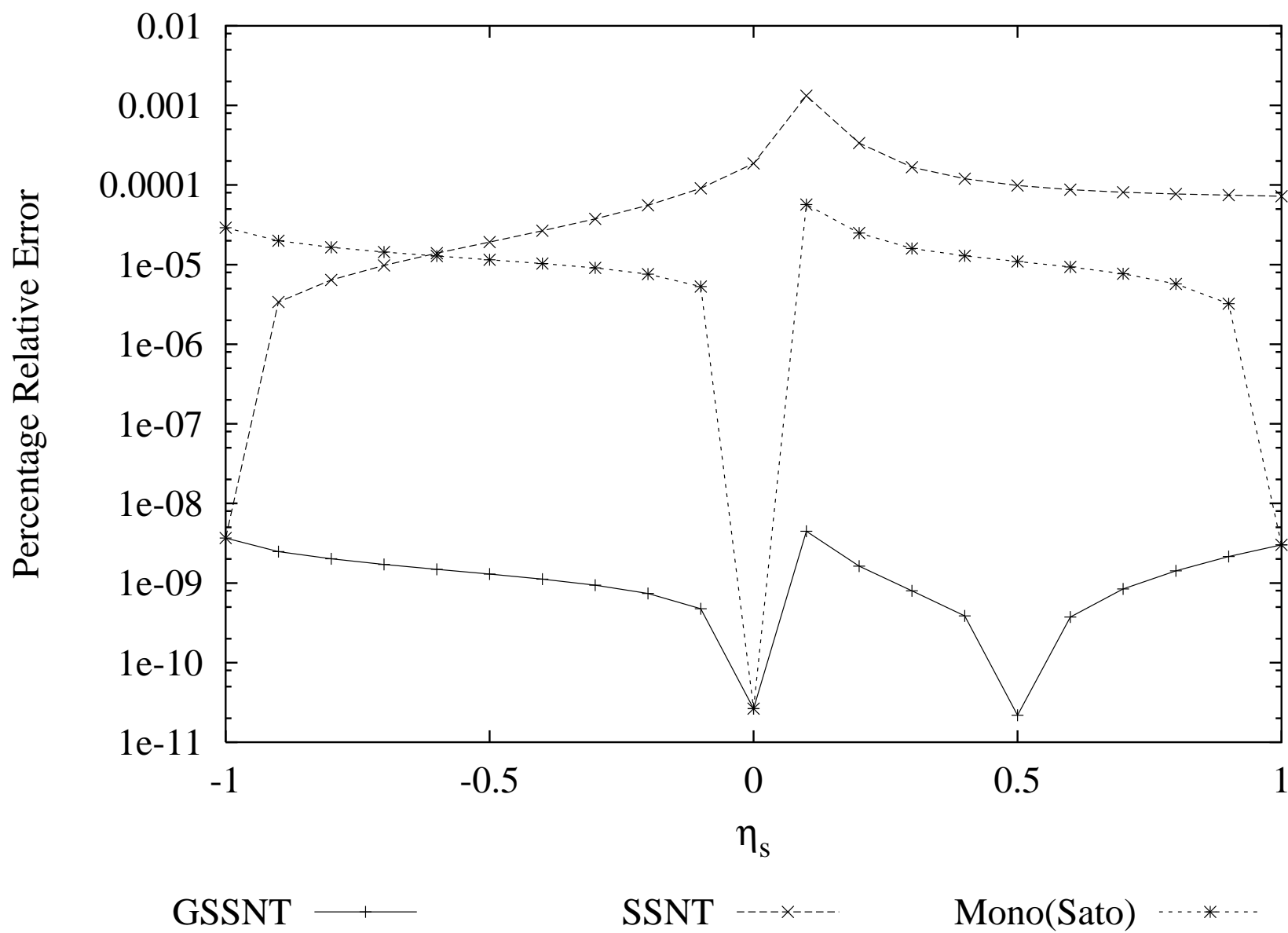


Figure 2:

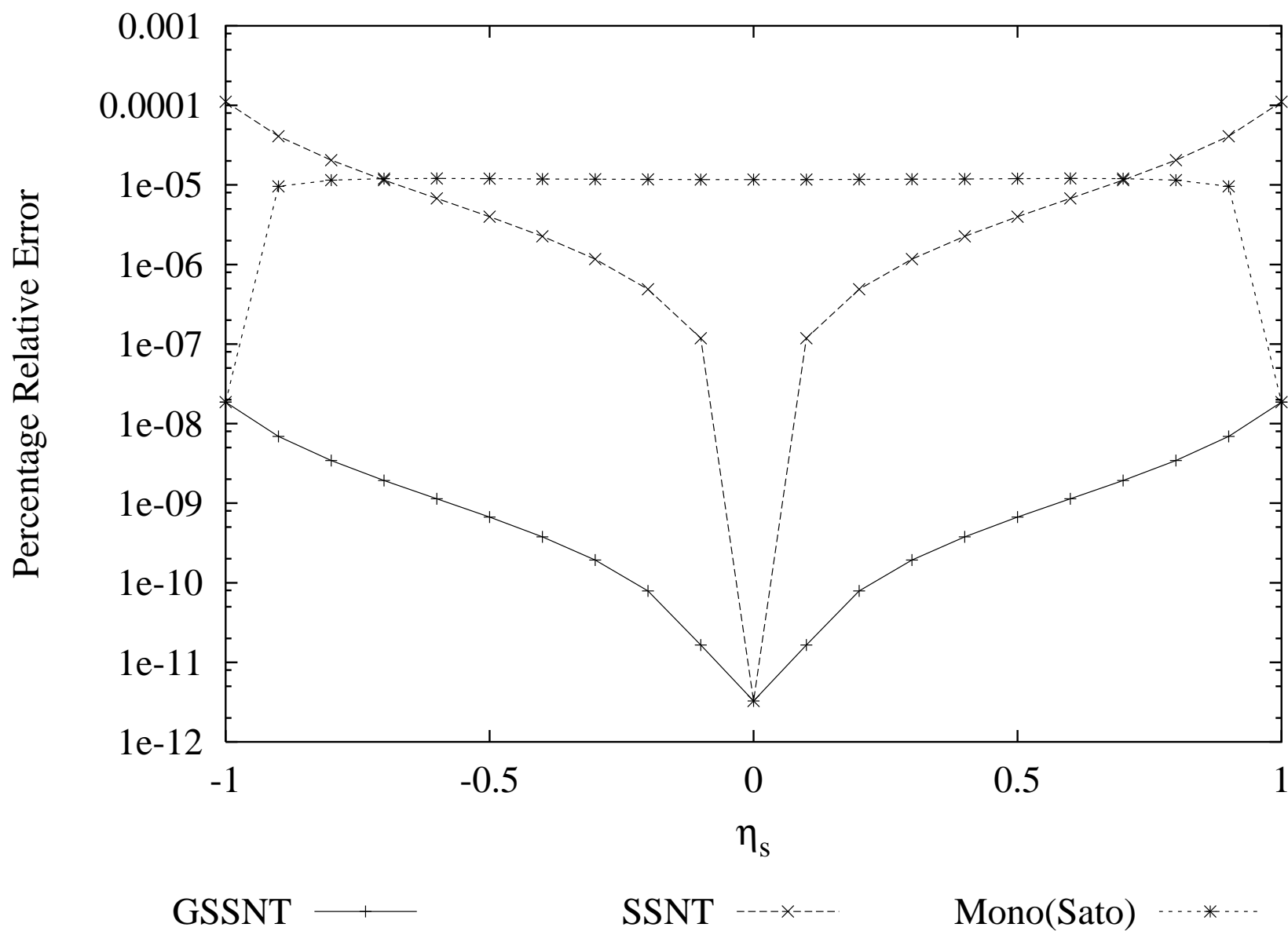
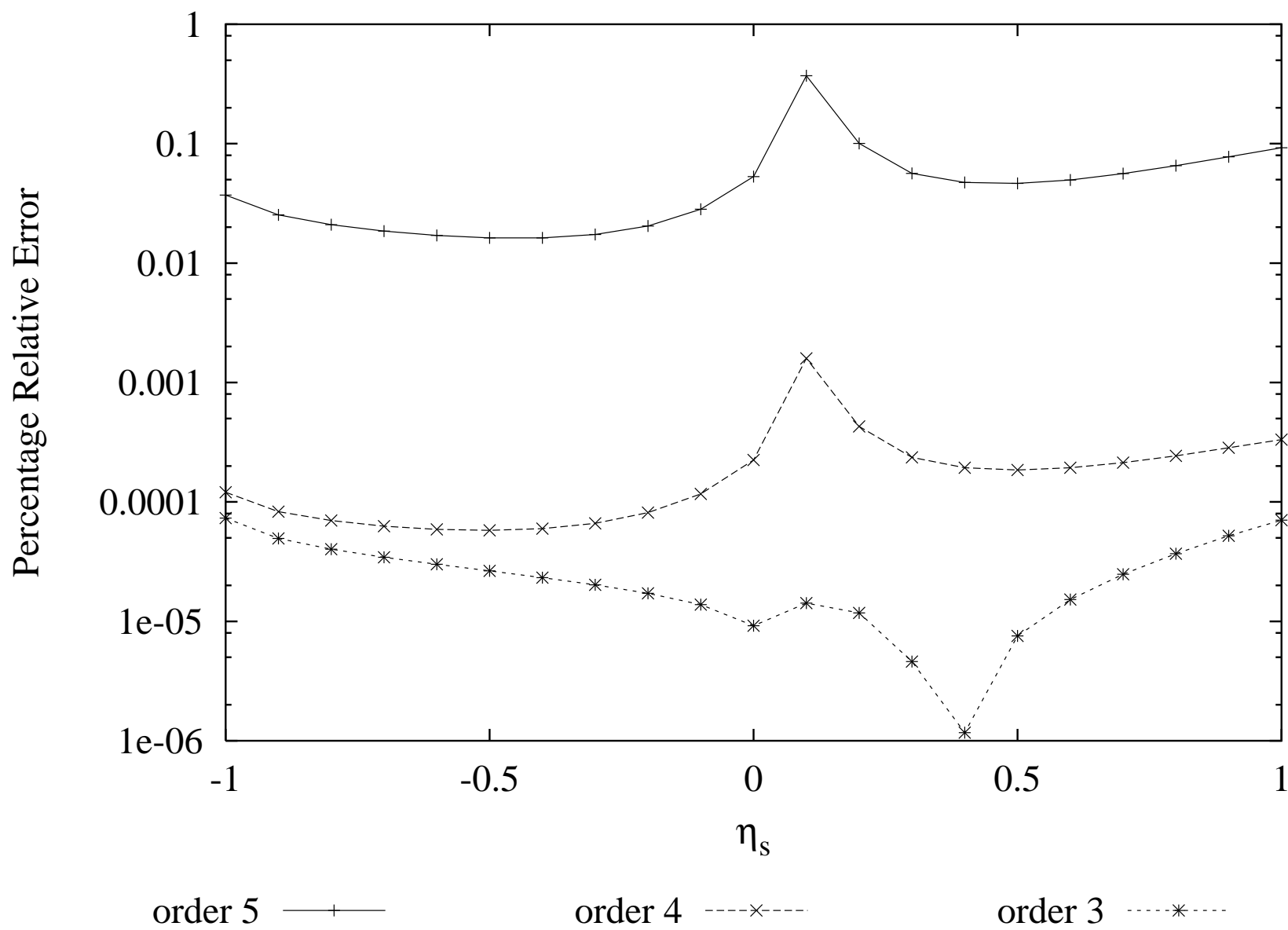


Figure 3:



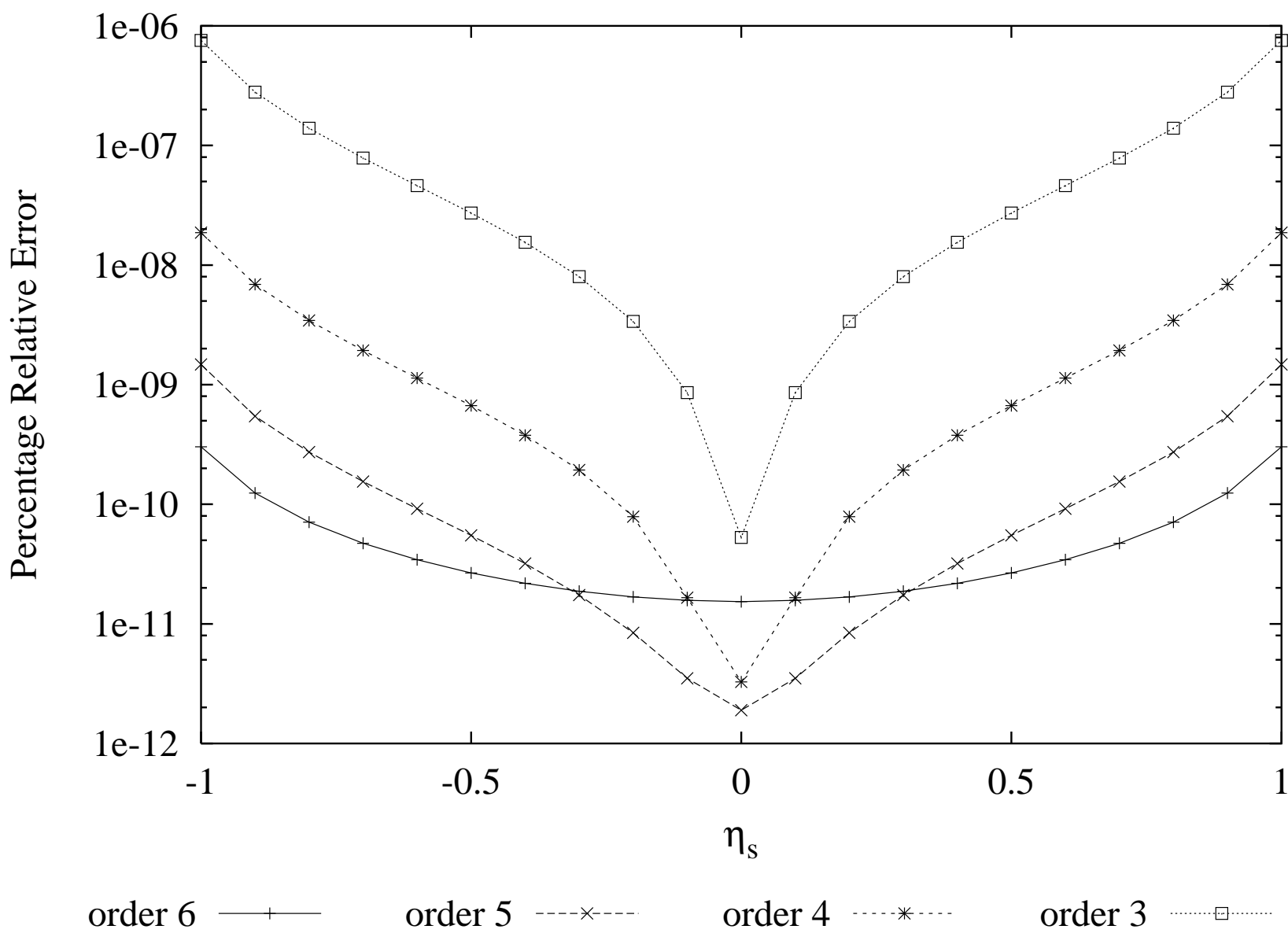


Figure 4: