# Adaptively Weighted Numerical Integration over Arbitrary Domains

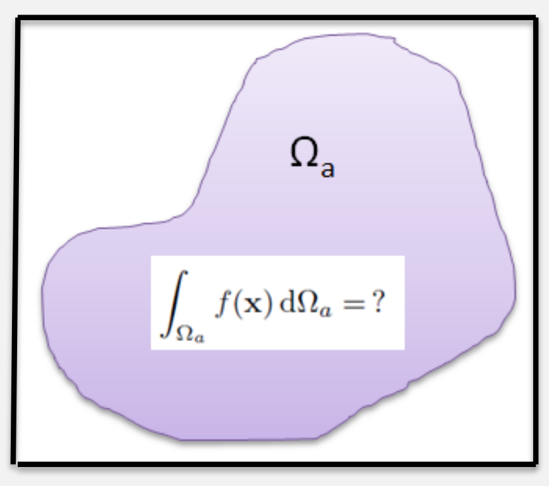
**Vaidyanathan Thiagarajan, Vadim Shapiro**

Spatial Automation Laboratory, University of Wisconsin-Madison, Madison, WI-53706

## ABSTRACT

In '*Adaptively Weighted Numerical Integration'* (AW), for a given set of basis functions, quadrature points, order and domain of integration, the quadrature weights are obtained by solving a system of suitable moment fitting equations in least square sense. The adaptivity of weights to the domain is ensured by incorporating a shape sensitivity term in formulating the moment fitting equations. Thus, the adaptivity of weights allow accurate integration over arbitrary 2D and 3D domains without excessive domain subdivision, which is useful in many applications and meshfree analysis in particular. Experimental results (in 2D) indicate that adaptively weighted integration compares favorably with more traditional approaches.

## OBJECTIVE



To develop an efficient method to numerically integrate a scalar valued continuous function $f(\boldsymbol{x})$ over an arbitrary 2D/3D domain $(\Omega_a)$ with no / minimal domain subdivision :

$$\boxed{\int_{\Omega_a} f(\mathrm{x})\,\mathrm{d}\Omega_a} \qquad (*)$$

## MOMENT FITTING EQUATIONS

- A quadrature in $R^d$ is an equation of the form $\sum_{k=1}^{n} w_k f(x_k) = \int_{\Omega_a} W(\boldsymbol{x}) f(\boldsymbol{x})\,\mathrm{d}\Omega_a$; where, $W(\boldsymbol{x})$ is a non-negative weighting function.
- The position $\{\boldsymbol{x}_i\}_{i=1}^{n}$ & weights $\{w_i\}_{i=1}^{n}$ of the quadrature points can be determined by solving the following set of non-linear moment fitting equations for a preselected set of basis functions $\{f_i\}_{i=1}^{m}$ :

$$\boxed{\underbrace{\begin{bmatrix} \int_{\Omega_a} W(\mathrm{x}) f_1(\mathrm{x})\mathrm{d}\Omega_a \\ \int_{\Omega_a} W(\mathrm{x}) f_2(\mathrm{x})\mathrm{d}\Omega_a \\ \cdots \\ \int_{\Omega_a} W(\mathrm{x}) f_m(\mathrm{x})\mathrm{d}\Omega_a \end{bmatrix}}_{\{L\}} = \underbrace{\begin{bmatrix} f_1(\mathrm{x_1}) & f_1(\mathrm{x_2}) & \cdots & f_1(\mathrm{x_n}) \\ f_2(\mathrm{x_1}) & f_2(\mathrm{x_2}) & \cdots & f_2(\mathrm{x_n}) \\ \cdots & \cdots & \cdots & \cdots \\ f_m(\mathrm{x_1}) & f_m(\mathrm{x_2}) & \cdots & f_m(\mathrm{x_n}) \end{bmatrix}}_{[A]} \cdot \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_n \end{bmatrix}}_{\{W\}}} \qquad (1)$$

- Such nonlinear equations can't be solved easily. However, fixing the position of quadrature points $\{\boldsymbol{x}_i\}_{i=1}^{n}$ turn the equations linear.
- Thus, quadrature weights can be obtained by solving $Eq.(1)$ in least square sense as :

$$\boxed{\{\mathbf{W}\}_{n\times 1} = [\mathbf{A}^{\dagger}]_{n\times m}\{\mathbf{L}\}_{m\times 1}} \qquad (2)$$

## APPROXIMATION OF MOMENTS

- **Shape Sensitivity Analysis (SSA)**

The integrals in the moment vector $\{\boldsymbol{L}\}$ (with $W(\boldsymbol{x}) = 1$) can be approximately computed over a polygonal/polyhedral domain $(\Omega_p)$ that is homeomorphic to the arbitrary domain $(\Omega_a)$ using Shape Sensitivity Analysis (SSA) [3] as follows:

$$\boxed{L_i = \int_{\Omega_a} f_i(\mathrm{x})d\Omega_a \approx \int_{\Omega_p} f_i(\mathrm{x})d\Omega_p + \int_{\Gamma_p} f_i(\mathrm{x})V_n(\mathrm{x})d\Gamma_p} \qquad (3)$$

where, $\boldsymbol{V}_n = t\,\widehat{\boldsymbol{v}}_n$ ; with $t \equiv$ time parameter & $\widehat{\boldsymbol{v}}_n \equiv$ Design Velocity

- **Gauss Divergence**

Further, integral of the basis functions over the polygonal/polyhedral domain $(\Omega_p)$ can be computed efficiently as a boundary integral by the application of 'Gauss Divergence Theorem' [1] as:

$$\boxed{\int_{\Omega_p} f_i(\mathrm{x})d\Omega_p = \sum_{k=1}^{n_f} \int_{\Gamma_p^k} \beta_x^i(\mathrm{x}) n_x^k d\Gamma_p^k} \qquad (4)$$

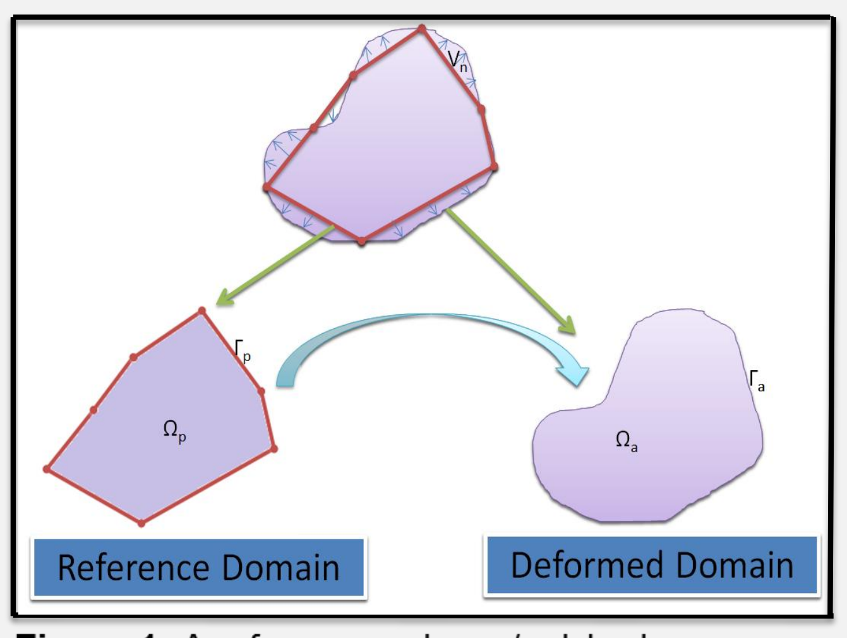


**Figure 1:** A reference polygon/polyhedron homeomorphically mapped to the arbitrary domain in order to approximate $\{L\}$ using SSA

where, $n_f$ is the number of edges/faces in the approximating polygon/polyhedron, $\beta_x{}^i$ is an integral (w.r.t. '$x$') that can be evaluated symbolically, and $n_x{}^k$ is the $x$-component of the normal to the '$k^{th}$' edge/face.

- **Final Expression**

Substituting $Eq.(4)$ in $Eq(3)$, we get the approximation to the moments on $\Omega_a$ to be an integral over the polygonal/polyhedral boundary $(\Gamma_p)$:

$$\boxed{L_i \approx \sum_{k=1}^{n_f} \int_{\Gamma_p^k} [\beta_x^i(\mathrm{x}) n_x^k + f_i(\mathrm{x})V_n(\mathrm{x})] d\Gamma_p^k} \qquad (5)$$

## 'AW' ALGORITHM

### Pre-Computations

- For the required order of integration, select an appropriate set of basis functions $\{f_i(\boldsymbol{x})\}$ (one simple choice is the bivariate $[x^p y^r]$ / trivariate $[x^p y^r z^s]$ polynomials).
- Compute [A†] symbolically for the chosen basis functions & desired order of integration and then write it to a file [one time computation].

### Main Computations $(O(n_f))$

- Choose the quadrature points $(\{\boldsymbol{x}_i\}_{i=1}^{n})$ appropriately [say using 'Scaled Cartesian product rule' – Fig. 2(a)].
- Choose a Polygon (in 2D) / Polyhedron (in 3D) that is a reasonable approximation & homeomorphic to the domain $(\Omega_a)$ (Fig. 2(b)).
- Compute the moments $(\{\boldsymbol{L}\})$ approximately using $Eq.(5)$ – an integral over the boundary of the polygon/polyhedron.

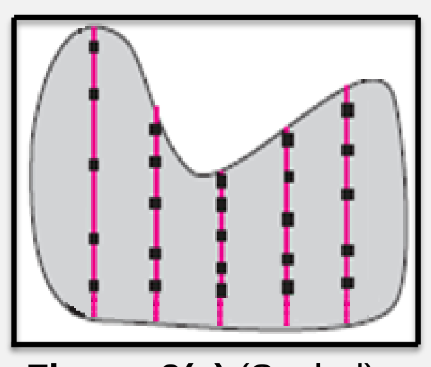


**Figure. 2(a)** (Scaled) Cartesian Product Rule

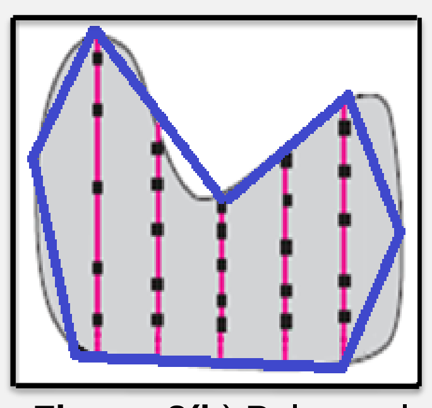


**Figure. 2(b)** Polygonal Approximation

- Read the appropriate [**A†**] matrix from the file and evaluate it at the selected quadrature points.
- With [**A†**] and $\{\boldsymbol{L}\}$ now known, solve the moment fitting equations $[Eq.(2)]$ for the (geometrically adaptive) weights $(w_k)$.
- Use the computed weights $(w_k)$ to approximate the integral over the arbitrary domain as :

$$\boxed{\int_{\Omega_a} f(\mathrm{x})\mathrm{d}\Omega_a \approx \sum_{k=1}^{n} w_k f(\mathrm{x_k})} \qquad (6)$$

## RESULTS AND DISCUSSION

- The algorithm for 2D domains was implemented in MATLAB and comparisons were made between 'AW', the (Scaled) Cartesian Product Gauss Quadrature ('C'), Direct Application of Shape Sensitivity ('SS') to $Eq.(*)$ and Geometrically Adaptive Method ('GA') [4].

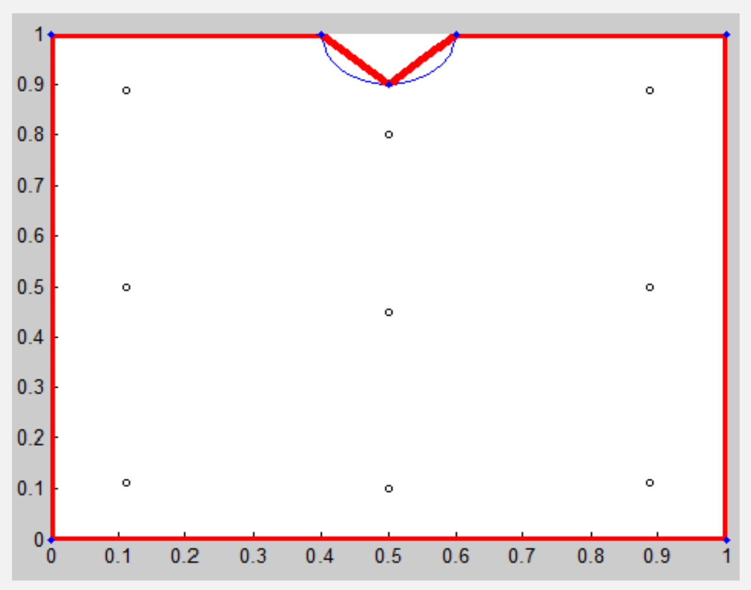


**Figure 3(a).** A 'notched' domain along with its polygonal approximation.

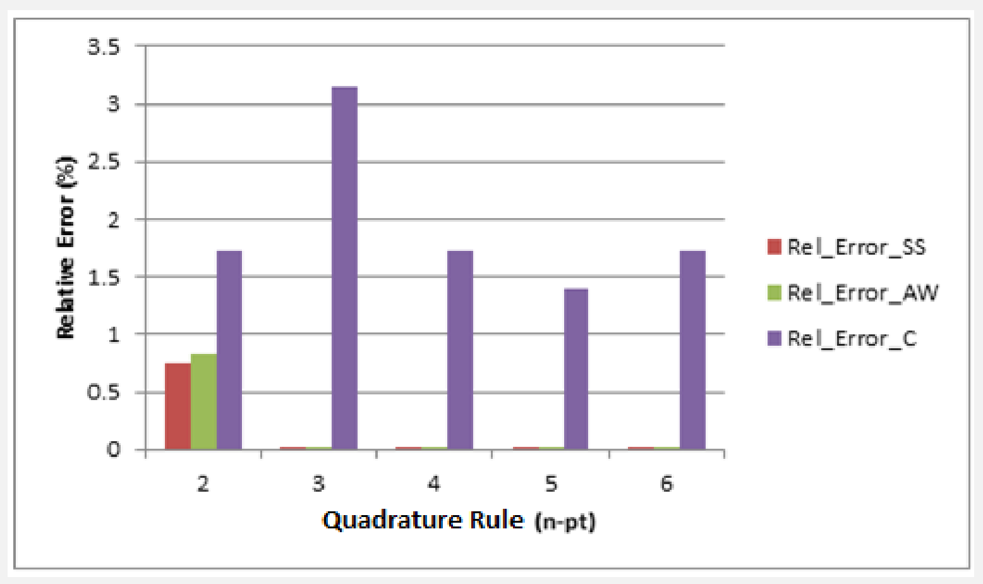


**Figure 3(b).** Relative error (in integral) comparison for various quadrature rule – 'notched' domain

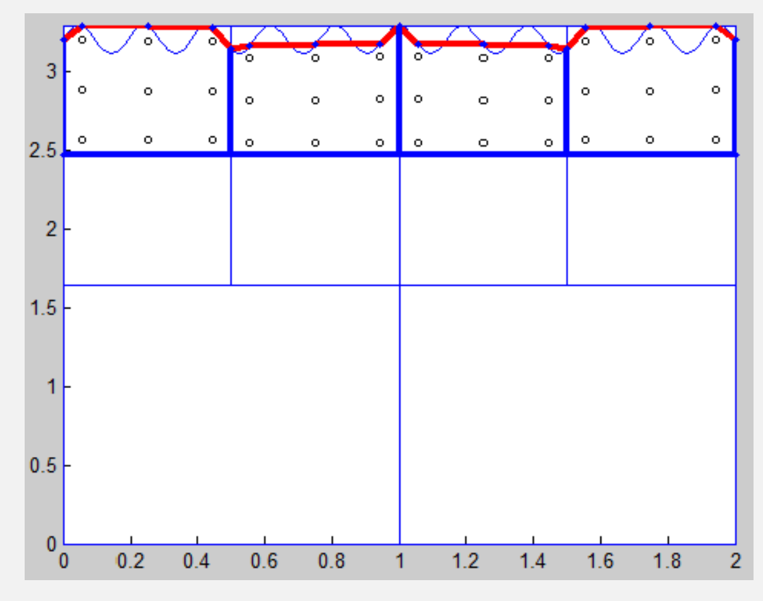


**Figure 4(a).** Quadtree decomposition of a 'wavy' domain along with the polygonal approximation of its boundary cells.

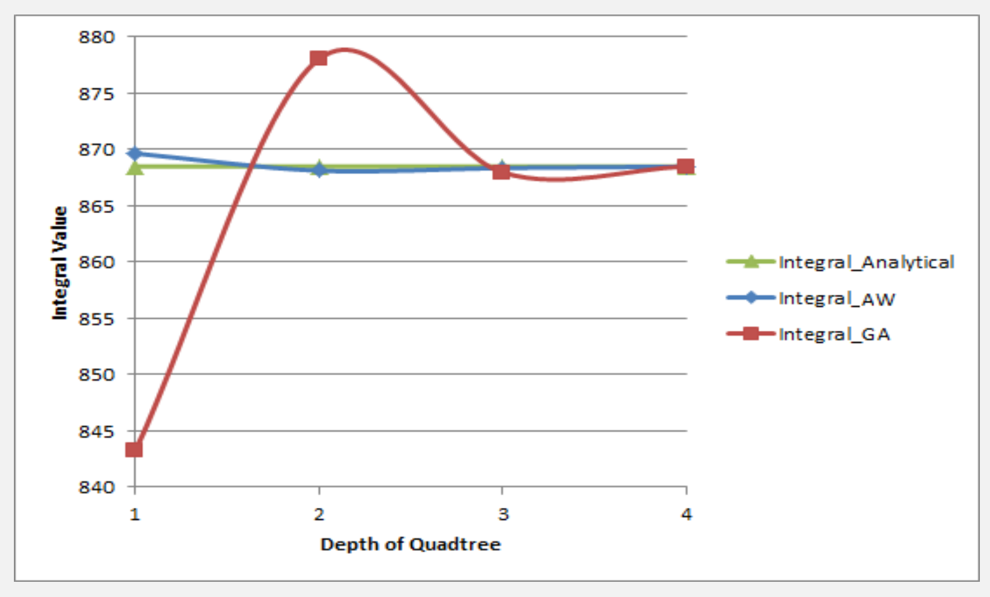


**Figure 4(b).** Integral value comparison for various levels of quadtree decomposition – 'wavy 'domain.

- An arbitrary function $'f = x^2y^2 + x^2y^3 + x^3 + 100x + 10y + 2'$ was integrated over a '*Notched*' domain (Fig. 3(a)) and a '*Wavy*' domain (Fig. 4(a)) using 'AW', 'SS', 'C' / 'GA' (3-pt quadrature) and MATLAB's symbolic toolbox.
- The results indicate that 'AW' requires *fewer function evaluations* than 'C' to achieve a desired accuracy (Fig. 3(b)).
- Also, 'AW', when used in quadtree boundary cells, *significantly decreases the number of subdivisions* required to resolve the geometry (for a prescribed accuracy) when compared to 'GA' (Fig. 4(b)).
- 'AW' is as accurate as 'SS'. However, unlike 'SS', 'AW' has the advantage of *not requiring integrand evaluation outside the domain* $(\Omega_a)$ *or design velocity computation inside the domain* $(\Omega_a)$.
- Apart from being *accurate* and *fast*, the AW method has several advantages including being *meshless (w.r.t geometric adaptivity)*, *representation independent*, *easy to implement* etc.

## REFERENCES

[1] Dasgupta G. Integration within polygonal finite elements. J. Aerosp. Eng., 2003.
[2] S. E. Mousavi and N. Sukumar. Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons. Computational Mechanics, 2011.
[3] K.K. Choi and N.H. Kim. Structural Sensitivity Analysis and Optimization I: Linear Systems. New York: Springer, 2005.
[4] Luft B., Shapiro V., and Tsukanov I. Geometrically adaptive numerical integration. In Proceedings of 2008 ACM Symposium on Solid and Physical Modeling, NY, 2008
[5] Thiagarajan V. and Shapiro V. Adaptively weighted numerical integration over arbitrary domains. Technical Report, Spatial Automation Laboratory, 2012.