BE for color graphics industry:
instrumentation and internet applications

by
Gautam Dasgupta and Elisabeth A. Malsch
Columbia University, New York, NY, USA
dasgupta@columbia.edu

Key words:
computer graphics, color interpolation, convex polygon, influence functions, medical imaging, online monitoring, perspective geometry, rational polynomial, shape factor, Wachspress macroelement.

Abstract

Fast rendering of responses from instrumented experiments, such as remote monitoring of microelectronic components, requires computationally fast and inexpensive color smoothing algorithms. Current web enabled implementations include VRML, cooperative design tools and online 3D game rendering engines. Constant color triangular patches make these tools unsuitable for high quality visualization of scientific data. An improvement is tessellation into quadrilaterals and subsequent color interpolation by Taig's isoparametric scheme. In pentagons, hexagons and all $n-$sided convex polygons, the only choice is Wachspress' rational polynomial interpolants. Unfortunately, computer programs are not widely available for that exact method. An alternative is to employ an available boundary element code that yields harmonic interpolations based on shape factors of convex polygons. The elliptic field operator is identified from perspective geometry by approximating the Wachspress' denominator. Computed color distributions are smooth in the domain and meet the boundary maxima/minima criterion. The technique sacrifices linearity between discrete nodal data. Consequent distortions in color are hardly noticeable after a linear masking around the edge. In monitoring remote experiments, this algorithm can be codified and then compiled for web browsers on any computer platform to display and update color graphics time series data.

April 18, 2001

submitted for review to Prof. J. Rencis for EABE (industrial application)

1 Introduction

Computer graphics literature,e.g., [5], introduces equivalent color schemes such as Red /Green /Blue (RGB) —used in computer monitors, Cyan /Magenta /Yellow /Black (CMYK) —used for printers and the more general Hue /Light /Saturation (HLS) measure for describing painting technics on images whose shapes, sizes and color distribution change with time. Hardly any commercially available rendering algorithm, e.g., `OpenGL`, or high-end software, e.g., Lightwave and Alias/Wavefront Maya, allows the user to consistently apply color gradient from RGB, CMYK or HLS values prescribed on boundary nodes of polygons beyond triangles ([2], [12]). Even with additional interior information, Figure 1 typifies the unevenness caused by piecewise linear interpolations. The proposed boundary element method alleviates this anomaly when nodal measurements preclude interior color data.

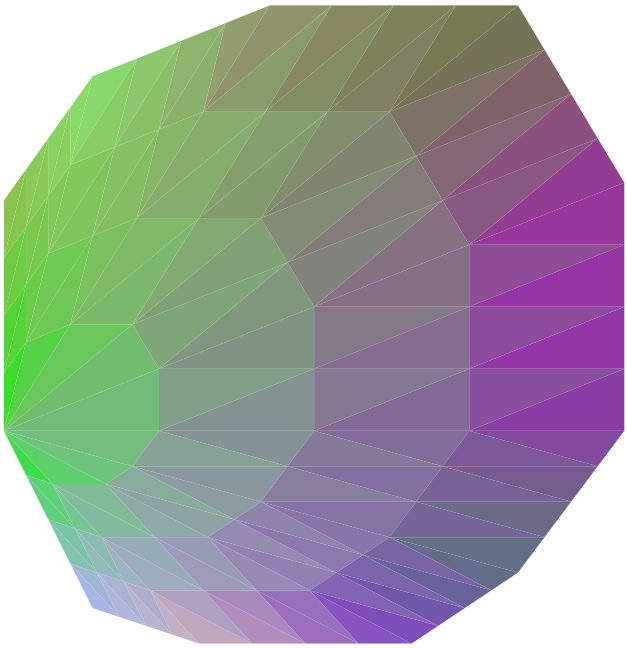

Figure 1: Triangular patches on an *endecagon* with 33 additional interior data

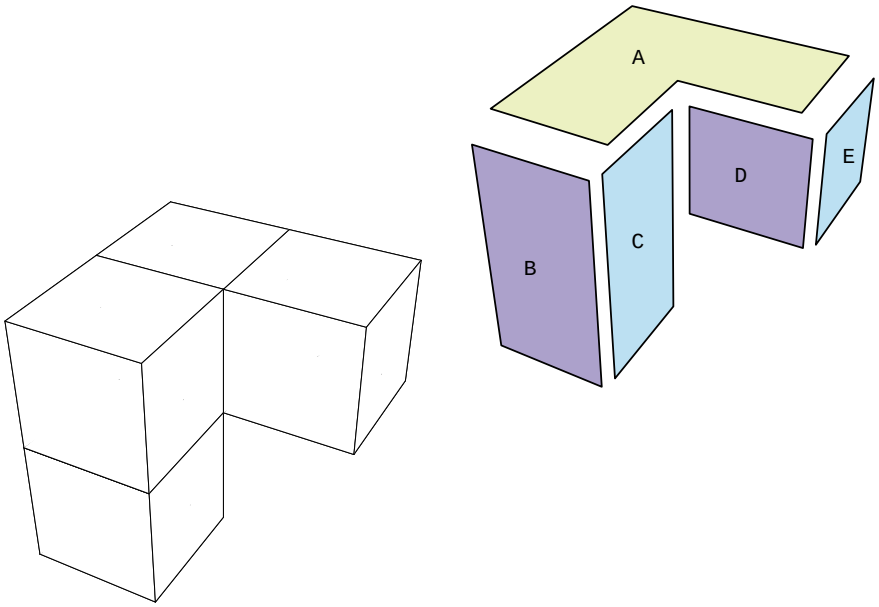

Figure 2: Concave object and convex faces

A three dimensional object is rendered by two dimensional shapes relative to the viewing angle shown in Figure 2. Concavity is resolved by combining convex objects or using a convex hull. Compatible color and their gradients across the artificial boundaries assure continuous lighting ([5]). Current graphics render engines discretize the screen polygons into constant field triangles ([12]). The discontinuity thus emerges are usually mitigated by time consuming anti-aliasing algorithms ([5]).

Sharp edges cause further distortion in color distribution which depends on the surface normal, relative distance from the light source, radiocity, light absorption and reflectivity. These parameters attain maximum and minimum along the boundary ([5]) thus amenable to boundary element formulations.

An optimized mesh, which tessellates with the minimum number of convex polygons, is more complex than the decomposed two dimensional wire frame ([6]). The time for image reconstruction is proportional to the number of convex subsets. The proposed boundary elements will interpolate the discrete nodal color of any convex $n-$gon vertices (Figure 3).

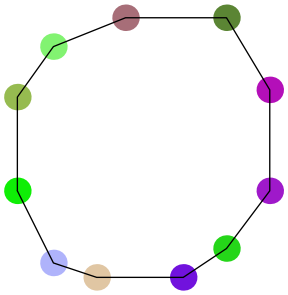

Figure 3: Color specified only at boundary nodes

Convex tessellation for interpolating color gradients is very computing intensive. The color resolution is dictated by the degree of tessellation. Unless the divisions are sufficiently small different discretization meshes result in different gradients. The only required tessellation is to define convex hulls as macro-elements [10]. This is optimal for two-dimensional images. Each convex domain can then be painted individually using only the prescribed boundary nodal color.

Continuous interpolants which satisfy the maximum, minimum principle insure that domain color changes smoothly and does not overshoot the boundary data. Such interpolants are conceived to satisfy partial differential field equations of the elliptic type. Consequently, the boundary element formulation can be used to construct an efficient and accurate coloring scheme. The resulting gradient is then defined on the entire picture and not as a collection of tessellated color patches. The results can be displayed at any desired resolution without any further boundary element computation. The color between two nodal points will not be distributed linearly, which is its only difference from Wachspress' formulation ([15]).

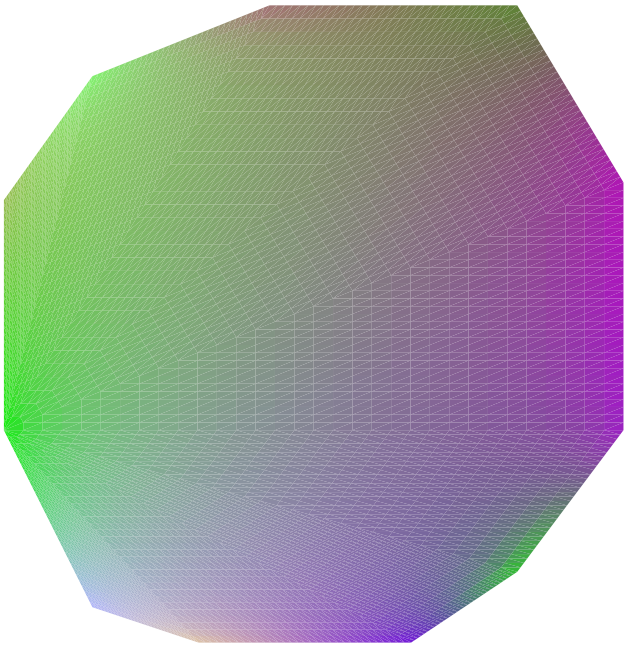

Figure 4: Wachspress' coloring from the nodal data of Figure 3

The smoothness of the color gradient according to the Wachspress scheme is in itself quite impressive (Figure 4). In this polygon with eleven sides, each interpolant is a closed-form rational polynomial, with a ninth order numerator and an eighth order denominator. The rendering time for the Wachspress method is equal to that needed to evaluate the rational polynomials only once. For dynamic and interactive display this minimal data set consists of nodal coordinates and their color values. Required color gradients are exactly reproducible at any resolution. Unfortunately no C or Fortran version of the program is commercially available for industrial applications. The boundary element scheme proposed here is anticipated to make a satisfactory approximation widely affordable.

Dictated by the perspective geometry a lower order interpolant, which contains a first order approximation of the singularity curve in Wachspress' projective geometry formulation, also reproduces smooth color. These influence functions can be used to identify a canonical Laplace's operator for the proposed boundary element calculations. This accurate and efficient scheme furnishes smooth color graphics. Linearity of color changes is a subjective determination and has not been enforced on the boundary.

In a number of medical applications concavity cannot be circumvented by convex tessellation. Modeling of human maxillo-facial frame, [14] was examined with radiological images furnishing the raw data (Figure 5). For morphometric computation the concavity cannot be ignored. The boundary element method is being tested to carry out the kinematic analysis. Subsequent coloring will require concave domains to be painted according to the computed values of second rank growth tensors.

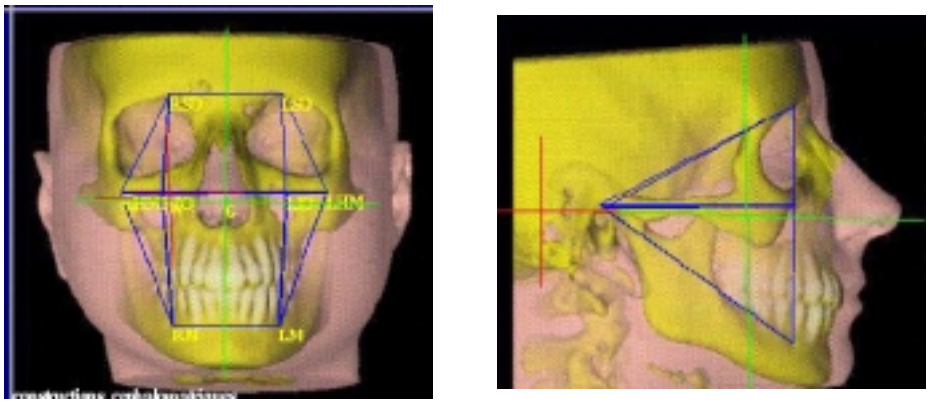

Figure 5: Treil Maxillo-Facial Frame: 3-D Concavity

2 Formulation

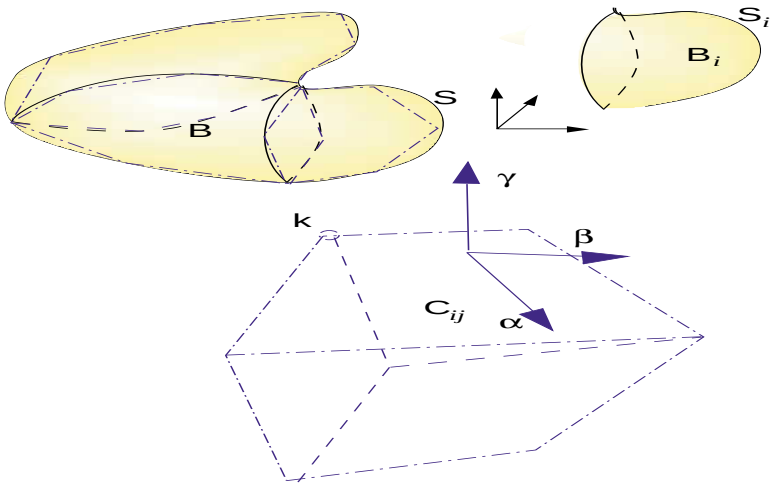

Figure 6: Convex tessellation: $\mathcal{B} \to \mathcal{B}_i$; $\mathcal{S}_i \to \mathcal{C}_{ij} \ni k$: node

Consider a three-dimensional graphic object $\mathcal{B}$ enclosed by its surface $\mathcal{S}$. In practice, only the color details on $\mathcal{S}$ need to be displayed. This surface in three-dimensions is approximated by a collection of plane patches. One such patch is a polygon, which can have any number of sides. Convex tessellation of the entire object $\mathcal{B}$ into $\bigcup_i \mathcal{B}_i$ guarantees that each three-dimension convex subpart $\mathcal{B}_i$ is enclosed by convex polygons $\mathcal{C}_{ij}$. The entire graphics object $\mathcal{B}$ is rendered by sequentially coloring an individual convex polygon $\mathcal{C}_{ij}$.

Let k be a node of $\mathcal{C}_{ij}$ whose normal is $\hat{\gamma}$. Let (α, β) define the coordinate system on $\mathcal{C}_{ij}$. The boundary element influence function $\mathcal{F}_{ijk}(\alpha, \beta)$ will be used as the coloring designator (`ColorFunction` in *Mathematica*) on $\mathcal{C}_{ij}$. The red, green and blue color values r_k, g_k, b_k will be prescribed at each node k. The (scaled) values of r, g, b lie bettween zero and unity without exception:

$$0 \leq r, g, b \leq 1 \tag{1}$$

To prevent any overshoot there is a crucial requirement:

$$1 \geq \mathcal{F}_{ijk}(\alpha, \beta) \geq 0; \quad (\alpha, \beta) \in \mathcal{C}_{ij} \tag{2}$$

Wachspress constructed $\mathcal{F}_{ijk}$ in rational polynomial forms for all convex $n-$sided polygons maintaining liearity along the boundary, consequently:

$$\mathcal{F}_{ijk}(\alpha_k, \beta_k) = 1, \, \mathcal{F}_{ijk}(\alpha_o, \beta_o) = 0, o \neq k; \tag{3}$$

$$(\alpha_k, \beta_k): \; k^{th}\text{nodal coordinates}; \; \mathcal{F}_{ijk}(\alpha_o, \beta_o) = \delta_{ko} : \text{Kronecker's } \delta \tag{4}$$

2.1 Interpolants: solutions of elliptic PDE

Equations (2) and (3) suggest that each interpolant $\mathcal{F}_{ijk}$, just like a coloring function, attains the maximum and minimum not within the interior of $\mathcal{C}_{ij}$ but on its boundary. In thermo-mechanics temperature interpolants must comply with the same requirement (any negative value of the temperature measured in Kelvin cannot be accepted). In the language of mathematical physics, $\mathcal{F}_{ijk}$ then satisfies some elliptic (not parabolic nor hyperbolic) partial differential equation. A convenient form is a linear operator $[\mathcal{L}]$:

$$\begin{aligned}[\mathcal{L}]\mathcal{F} = 0; \quad & \|\mathcal{M}\| > 0; \quad \mathcal{M} : \text{the characteristic matrix of operator } [\mathcal{L}]; \\ & \|x\| \text{ is any spectral norm}\end{aligned} \tag{5}$$

An choice for $[\mathcal{L}]$ could be the potential equation for anisotropic media:

$$[\mathcal{L}] = f_1(\alpha,\beta)\frac{\partial^2}{\partial\alpha^2} + f_2(\alpha,\beta)\frac{\partial^2}{\partial\beta^2}; f_1, f_2 > 0, (\alpha,\beta) \in \mathcal{C}_{ij} \tag{6}$$

2.2 Coordinate tansformation and interpolation

Taig's concept of isoparametric transformation [13] has empowed the finite element method to seek solutions beyond the linear interpolants on triangles. He brilliantly postulates the same interpolants for responses and coordinates. On a convex quadrilateral, the Wachspress response interpolants from projective geometry were found to be identical with the perspective transformations (used in architectural drawings) on a unit square, *vide* [11] for details and examples. It is remarkable that for parallelograms, which results from an affine transformation, Taig's and Wachspress' interpolants are the same. The salient features of the affine transformation and the perspective transformation with homogeneous coordinates are summarized below.

2.2.1 Affine Transformation

The conventional engineering drawings are generated by scaling, rotating and translating polygon like objects. This is achieved by affine transformation, [11]. The general two-dimensional transformation of (x, y) to (α, β) is carried out according to following matrix multiplication:

$$\begin{Bmatrix}\alpha\\ \beta\end{Bmatrix} = \begin{bmatrix}a_{11} & c\\ d & a_{22}\end{bmatrix}\begin{Bmatrix}x\\ y\end{Bmatrix}; \quad (x,y): \text{ object}; \quad (\alpha,\beta): \text{ image}; \tag{7}$$

The elements a, b, c and d dictate shape changes between an object and its image. For nonvanishing image area the matrix in equation (7) is nonsingular:

$$\det \begin{bmatrix} a_{11} & c \\ d & a_{22} \end{bmatrix} \neq 0 \tag{8}$$

Traditionally, the coordinates $\begin{Bmatrix} x \\ y \end{Bmatrix}$ and $\begin{Bmatrix} \alpha \\ \beta \end{Bmatrix}$ are stored as row vectors $< x, y >$ and $< \alpha, \beta >$, respectively, in the computer graphics literature. The same convention is followed in C and *Mathematica* programming languages, and here we shall follow the same. Consequently:

$$< \alpha, \beta >=< x, y > \begin{bmatrix} a_{11} & d \\ c & a_{22} \end{bmatrix} \tag{9}$$

The affine transformation in equation (9) maps a system of parallel lines into another set of parallel lines. In order to transform the unit square into a trapezoid, the following homogeneous coordinate system is introduced, *vide* [11] for details.

2.2.2 Homogeneous coordinates

A point (x, y) is augmented to three-dimensions as $(x, y, 1)$. The third coordinate in the transformation is the homogenizing parameter h. The result (α, β, h) leads to the image coordinates (α^*, β^*) as follows:

$$< \alpha, \beta, h >=< x, y, 1 > \begin{bmatrix} a_{11} & a_{12} & p \\ a_{21} & a_{22} & q \\ a_{31} & a_{32} & s \end{bmatrix}; \quad \alpha^* = \frac{\alpha}{h}, \quad \beta^* = \frac{\beta}{h} \tag{10}$$

The image point in terms of the nine constants: a_{ij}, p, q, s becomes:

$$\alpha^* = \frac{a_{11}x + a_{21}y + a_{31}}{px + qy + s}; \quad \beta^* = \frac{a_{12}x + a_{22}y + a_{32}}{px + qy + s} \tag{11}$$

perspective constants p, q, s in the denominator indicate singularity

This rational polynomial form for coordinate transformation suggests that transformation of a square into an arbitrary quadrilateral is best formulated by the Padé form introduced by Wachspress as early as in 1975. These interpolants are smooth and attain maximum/minimum on the boundary.

Equation 11, which appears to be a nonlinear transformation from (x, y), maps a straightline into another exact straightline. The perspective transformation coincides with the Wachpress' interpolants for all convex quadrilaterals. This is a remarkable result.

The elliptic field operator $[\mathcal{L}]$ in equation (12) that governs the transformation is yet to be determined. Hence the associated boundary element method cannot be directly employed. On the other hand equation (11) clearly indicated the possible form for the denominator in the singular fundamental solution for the field operator $[\mathcal{L}]$ to be used in the proposed boundary element formulation. This facilitates the construction of the boundary element solutions via influence functions.

2.2.3 Isoparametric vs. Wachspress' shapes and BE influence functions

Let us consider a squrae whose four corners are (x_k, y_k), k=1, ... 4. The corresponding (α_k, β_k), k=1, ... 4, from equation (11) always contains Wachspress' exact denominator. When at least two sides are parallel Taig's isoparametric elements have been verified to yield the exact results. His heuristic ingeneous approximation is commendable. The denominator in equation (11) becomes a constant. However, for general quadrilaterals when no two sides are parallel, isopametric results are not in the Padé form. They are square roots of polynomials to the contrary. This establishes that there exists no algebraic geometrical (rigorous) basis for the isoparametric formulation.

We would like to capture interpolants close to the exact Wachspress solution, which are valid for all convex $n-$sided polygons rather than the one produced by Taig's isoparametric form, which is limited to quadrilaterals. These interpolants will be obtained as solutions to elliptic partial differential equations. The associated boundary element influence functions will contain the denominator polynomial of equation (11). We shall selecte the popular field equation for potentials to govern the (r, g, b) color in the form:

$$[\mathcal{L}] \cong \frac{\partial^2}{\partial x^2} + c_o \frac{\partial^2}{\partial y^2} = 0; c_o > 0; \tag{12}$$

Perception of a chromatic domain depend upon the shape factor. This is commonly represented by a nondimensional parameter related to the ratio of principal second area moments (commonly termed as moments of inertia). From the mechanics viewpoint we are interested only in the kinematical

aspect of color interpolants since no constitutive relation is involved. The constant c_o in equation (12) depends on the shape factor of the element. Experience has taught that $c_o = 1$ tends to distribute red, green and blue colors rather unevenly for irregular polygons.

The advantage of equation (12) is that existing boundary element codes, e.g., BEASY [1], can be readily employed to solve any coloring problem. Computation of the key positive constant c_o is based on the denominator polynomial in equation (11). It has been verified that the distribution of sources for the Green's function of equation (12) along the straight line $px + qy + s = 0$ yields the *exact* interpolations on quadrilaterals. This established the close connection between the Wachpress' projective geometry approach, homogenous coordinate transformation of the perspective geometry and the boundary element formulation. An extension on convex domains with any number of sides will be based on this successful quadrilateral model. The interpolants will deviate from Wachspress' exact linear form on the boundary thus may cause over or under shooting ($0 > r, g, b > 1$). A linear masking around the perimeter alleviates the anomaly. However, the color gradient error in the interior is hardly significant.

2.3 Construction of interpolants

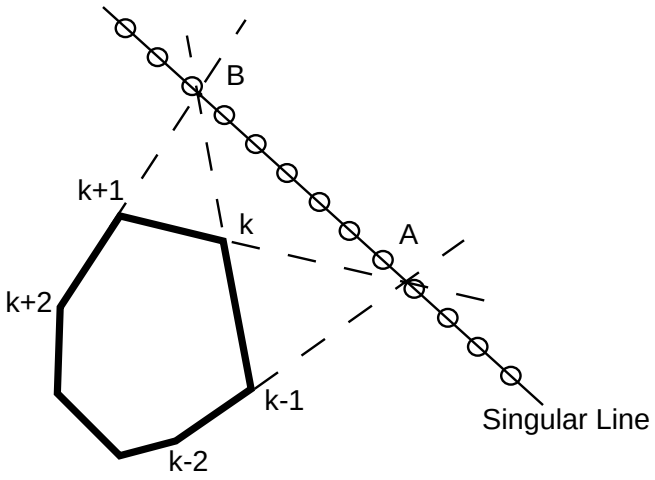

Figure 7: Approximates source line on a septagon

Consider a convex polygon in Figure 7. Select a node k. Connect the lines

through nodes $k-1, k-2$ and $k, k+1$, Their intersection gives one EIP (external intersection point), which is dented by A. Now joining the nodes $k, k-1$ and $k+1, k+2$ obtain the second EIP shown as B.

Instead of taking the algebraic curve through all EIPs as needed in the exact formulation of Wachspress, here we limit the singular sources only btween A and B to obtain the interpolant $\mathcal{F}_{ijk}$. From a number of numerical experiment it can be concluded that such anomaly have secondary effects on the visual effect in color graphics.

The algebraic expression for the first interpolant $\mathcal{F}_{ij1}(k=1)$ is selected in the following form to vanish along all sides not through node-1:

$$F_{ij1}(\alpha,\beta) = c\frac{\prod_m \ell_m(\alpha,\beta)}{\ell_o(\alpha,\beta)}, \quad c: \text{ set to assigned unit value at node-1} \tag{13}$$

$$\ell_m: \text{ not through node-1}$$

$$\ell_m = 1+\eta_m\alpha+\zeta_m\beta; \quad \ell_m = 0: \text{ for side } m; \tag{14}$$

$$\ell_o = 1+p\alpha+q\beta; \quad \ell_o = 0: \text{line AB} \tag{15}$$

The field operator, $[\mathcal{L}]$ of equation (6) will yield $\mathcal{F}$, which vanishes along all sides not containing the node-1. The following form is assumed:

$$[\mathcal{L}] = \frac{\partial^2}{\partial\alpha^2} + \mu(\alpha,\beta)\frac{\partial^2}{\partial\beta^2}, \quad \text{where} \tag{16}$$

$$\mu(\alpha,\beta): \text{ is the } \textit{yet to be determined} \text{ positive function from:}$$

$$[\mathcal{L}]\,\mathcal{F}_{ij1} = 0, (\alpha,\beta) \in \mathcal{C}_{ij} \tag{17}$$

From equation (16) a lengthy algebraic expression for $\mu(\alpha,\beta)$:

$$\mu(\alpha,\beta) = -\frac{\frac{\partial^2\mathcal{F}}{\partial\alpha^2}}{\frac{\partial^2\mathcal{F}}{\partial\beta^2}} \tag{18}$$

makes $[\mathcal{L}]$ intractable with a boundary element code. Fortunately, a crude first order approximation:

$$\mu(\alpha,\beta) \cong \mu(\alpha_o,\beta_o) \quad \text{where } (\alpha_o,\beta_o) \text{ is the centroid of } \mathcal{C}_{ij} \tag{19}$$

has shown to yield satisfactory color distribution.

The value of c_o will depend upon the choice of the first node. Hence for objectivity average:

$$c_o = \frac{1}{n}\sum_k c_o(k), \text{ for } n-\text{gon} \tag{20}$$

is selected. For all convex polygons c_o is positive.

Computer algebra, see Appendix-I, has enormously facilitated the numerical evaluation of:

$$c_o(k) = -\frac{\frac{\partial^2 \mathcal{F}}{\partial \alpha^2}}{\frac{\partial^2 \mathcal{F}}{\partial \beta^2}}\Big|_{\alpha,\beta=0} \quad \text{reducing to: } -\frac{p^2 - p\sum \eta_k + \sum\limits_{k\neq m} \eta_k \eta_m}{q^2 - q\sum \zeta_k + \sum\limits_{k\neq m} \zeta_k \zeta_m} \tag{21}$$

This simplification leads to $[\mathcal{L}]$ which can be easily solved by a boundary element code, such as BEASY, [1]. This color interpolation is suitable for rendering high quality scientific data for research industries.

An alternative procedure is to implement a web-enabled approximate form of the boundary element solution when some accuracy could be sacrified in the interest of speed. Instead of using the logartihm function to construct the boundary element system matries $[G]$ and $[H]$ in $[G]\{u\}=[H]\{\tau\}$ for color values $\{u\}$ and color gradients $\{\tau\}$, the influence functions $\mathcal{F}_{ijk}, k = 2, 3, \ldots$ can be obtained from $\mathcal{F}_{ij1}$ as elaborated below.

2.4 Interdependency of influence functions

The compact color data for a patch $\mathcal{C}_{ij}$ can be transmitted as:

$$((\alpha_k, \beta_k), (r_k, g_k, b_k)), \text{ for each node } k \text{ of } \mathcal{C}_{ij} \tag{22}$$

The information size is considerably smaller than the coordinate and color data for each point on the screen. The client application reconstructs the color image on $\mathcal{C}_{ij}$ according to:

$$(r(\alpha,\beta), g(\alpha,\beta), b(\alpha,\beta)) = \sum_k^n (r_k, g_k, b_k)\mathcal{F}_{ijk}(\alpha,\beta); \quad \mathcal{C}_{ij} : n-\text{gon} \tag{23}$$

Heuristically, we select a set of higher order (quadratic, cubic, etc.) profiles to guide color smoothing. An illustration of a pentagonal patch follows.

On the patch $\mathcal{C}_{ij}$ in Figure 8 we start with interpolant $\mathcal{F}_{ij1}(\alpha,\beta)$ from equation (13). The remaining four interpolants will exactly reproduce the mandatory lower order linear fields (*vide* Iron's patch test[7]):

$$1,\alpha,\beta \quad \text{by} \sum_k \mathcal{F}_{ijk}(\alpha,\beta) = 1 - (\text{independent of } \alpha,\beta)$$
$$\text{and} \qquad \sum_k \alpha_k \mathcal{F}_{ijk}(\alpha,\beta) = \alpha; \quad \sum_k \beta_k \mathcal{F}_{ijk}(\alpha,\beta) = \beta \tag{24}$$

The fourth quadratic profile is to be selected to exactly satisfy:

$$\alpha^2 \text{ or } \alpha\beta \text{ or } \beta^2 \tag{25}$$

There is no hard and fast rule to ascertain the superiority of one over the other. Let us arbitrarily select the mixed term $\alpha\beta$ in equation (25) thus:

$$\sum_k \alpha_k \beta_k \mathcal{F}_{ijk}(\alpha,\beta) = \alpha\beta \tag{26}$$

Now, $\mathcal{F}_{ijk}, k = 2,4$ can be uniquely determined from $\mathcal{F}_{ij1}$ by solving the linear system equation (24). This processs carries over the first order singularity from $\mathcal{F}_{ij1}$ to other interpolants $\mathcal{F}_{ijk}, k \neq 1$, intact. Wachspress' exact solution can be obtained when $\mathcal{C}_{ij}$ is a convex quadrilateral.

3 Example – Pentagon – in a generic (x,y) frame

The procedure to calculate influence functions $\mathcal{F}_{ijk}$ for a generic pentagon follows. The *Mathematica* steps are illustrated with five nodes in Figure 8:

```
n5={{3, 2}, {2, 2.5}, {0, 1}, {0, 0}, {1.5, 0}}
```
(27)

It is convenient to select the centroid as the origin. Convex domains always contain their centroids as internal points. The centroidal coordinates are:

```
q5={{1.627, 0.873}, {0.627,  1.373}, {-1.372, -0.127},
    {-1.373, -1.127}, {0.127, -1.127}}
```
(28)

The curve through the external intersection points, Figure 8, represents the *exact* denominator polynomial that appears in the Wachspress formulation.

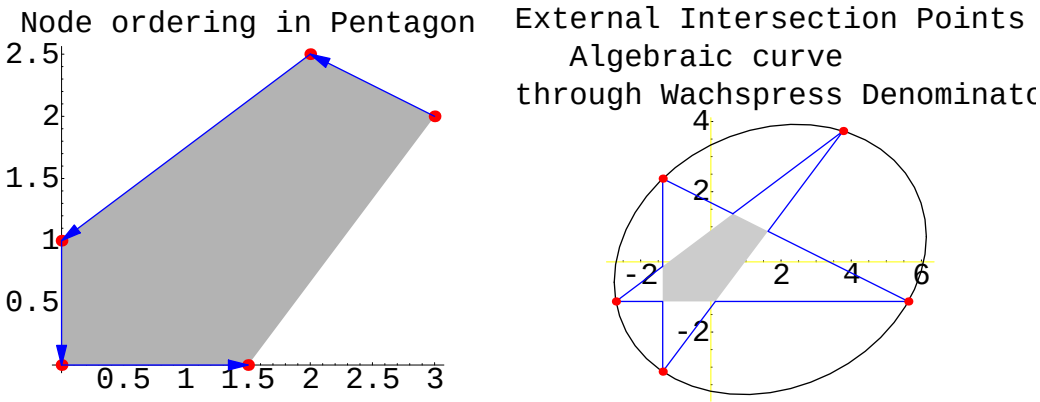

Figure 8: Pentagonal element with Wachspress denominator curve

We shall demonstrate the construction of the interpolant associated with node number one. The two external intersection points to be used are:

```
{{5.62745, -1.12745}, {3.77031, 3.72969}}
```
(29)

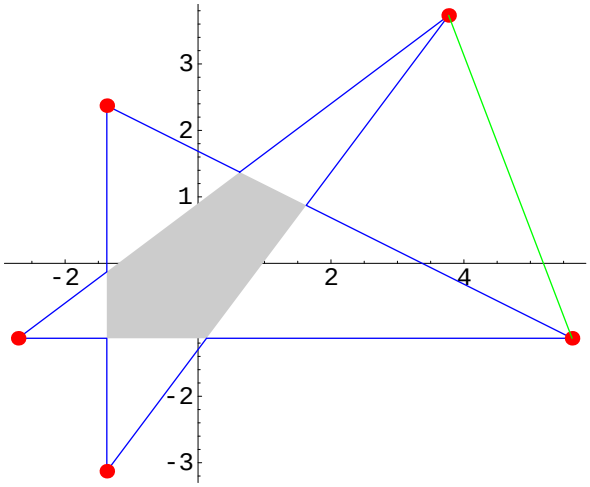

Figure 9: Singularity Line for boundary element sources

The straight line through the points in equation (29) represents singularity:

$$s_o = 1 - 0.192442\,x - 0.0735808\,y; \quad s_o = 0 : \text{ line of singularity} \tag{30}$$

instead of the exact ellipse in Figure 8:

$$5.032 + 1.03\,x - 0.31\,x^2 - 0.145\,y + 0.119\,x\,y - 0.409\,y^2 = 0 \tag{31}$$

The approximated algebraic form of the first interpolant $\mathcal{F}_{ij1}(x, y)$ becomes:

$$\mathcal{F}_{ij1}(x,y) = c\frac{\ell_2\ell_3\ell_4}{s_0}; \quad \ell_i: \text{ line through nodes: } i, i+1; \; c: \text{ scaling constant} \tag{32}$$

$$\ell_2 = 1 + 0.831522\,x - 1.1087\,y, \quad \ell_3 = 1.37255 + x, \quad \ell_4 = 1.12745 + y \tag{33}$$

The constant $c = \frac{1}{13.3555}$ in equation (32) yields unit nodal value leading to :

$$\mathcal{F}_{ij1}(x,y) = \frac{(1 + 0.831522\,x - 1.1087\,y)(1.37255 + x)(1.12745 + y)}{13.3555(1 - 0.192442\,x - 0.0735808\,y)} \tag{34}$$

In the elliptic field equation for boundary element formulation, equation (12), the unknown scaling constant is obtained as 0.948523, thus:

$$[\mathcal{L}] = \frac{\partial^2 u}{\partial x^2} + 0.948523\frac{\partial^2 u}{\partial y^2} \tag{35}$$

This individual value is not objective since it started with node numbered 1. As nodal numbers rotate right the coefficient drastically changes to:

```
{0.948523, -0.202463, -1.30343, 11.5767, 0.499167}
```
(36)

The average value is selected to govern the boundary element calculations according to the following elliptic partial differential equation:

$$\frac{\partial^2 u}{\partial x^2} + 2.3037\,\frac{\partial^2 u}{\partial y^2} = 0 \tag{37}$$

For convex domains the average constant is always positive. The spread, as shown in equation (36), and especially the negative values indicate the error in the first order approximation.

The coloring scheme proceeds by painting the pentagonal region in Figure 8 according to an available boundary element code where the field equation, equation (37), generates the Green's function. The prescribed nodal colors furnish the boundary data. The result is shown in Figure 10.

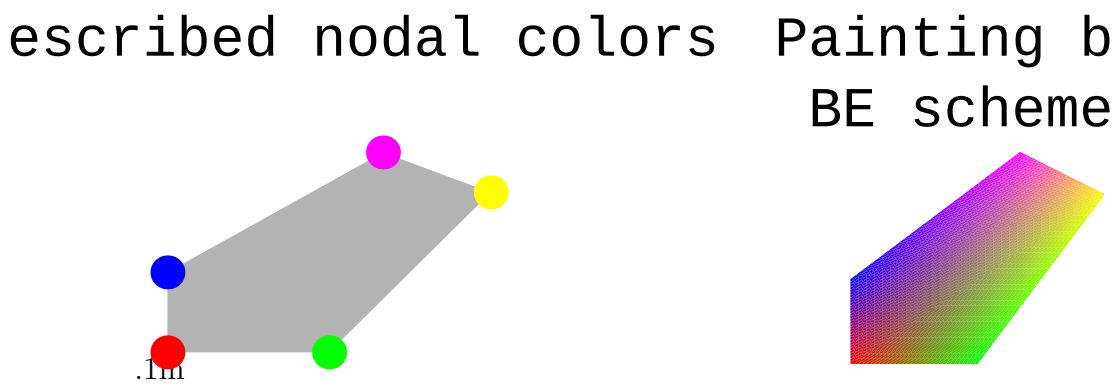

Figure 10: Coloring according to the BE formulation

An algebraic approximation of the boundary element solution can be obtained from:

$$\mathcal{F}_{ij1}(x,y) = \frac{(1+0.831522\,x-1.1087\,y)(1.37255+x)(1.12745+y)}{13.3555\ (1-0.192442\,x-0.0735808\,y)} \tag{38}$$

Functions $\mathcal{F}_{ij2}(x,y), \mathcal{F}_{ij3}(x,y), \mathcal{F}_{ij4}(x,y)$ and $\mathcal{F}_{ij5}(x,y)$ are sought to exactly interpolate a constant, linear and a mixed quadratic term:

$$\{1, x, y, x\,y\} \tag{39}$$

using the *Mathematica* program presented in Appendix-II. The results are:

$$\begin{aligned}
\mathcal{F}_{ij2}(x,y) &= (-0.885741+0.264957\,x+0.663206\,x^2-1.46833\,y-0.262399\,x\,y\\
&\quad +0.588235\,x^2\,y-0.605536\,y^2-0.441176\,x\,y^2)/deno\\
\mathcal{F}_{ij3}(x,y) &= (-2.4401+2.34371\,x-0.928489\,x^2-1.36148\,y+2.34958\,x\,y\\
&\quad -0.823529\,x^2\,y+0.712034\,y^2+0.240196\,x\,y^2)/deno\\
\mathcal{F}_{ij4}(x,y) &= (1.10124+0.354496\,x+0.118128\,x^2+1.03804\,y-3.03874\,x\,y\\
&\quad +0.696078\,x^2\,y-0.321799\,y^2+0.0441176\,x\,y^2)/deno\\
\mathcal{F}_{ij5}(x,y) &= (-2.36966-1.02384\,x+0.511918\,x^2+2.04062\,y+1.29835\,x\,y\\
&\quad -0.137255\,x^2\,y-0.376778\,y^2-0.27451\,x\,y^2)/deno\\
deno &= -5.19637+1.\,x+0.382353\,y
\end{aligned} \tag{40}$$

The denominator is proportional to that in the starting interpolant.

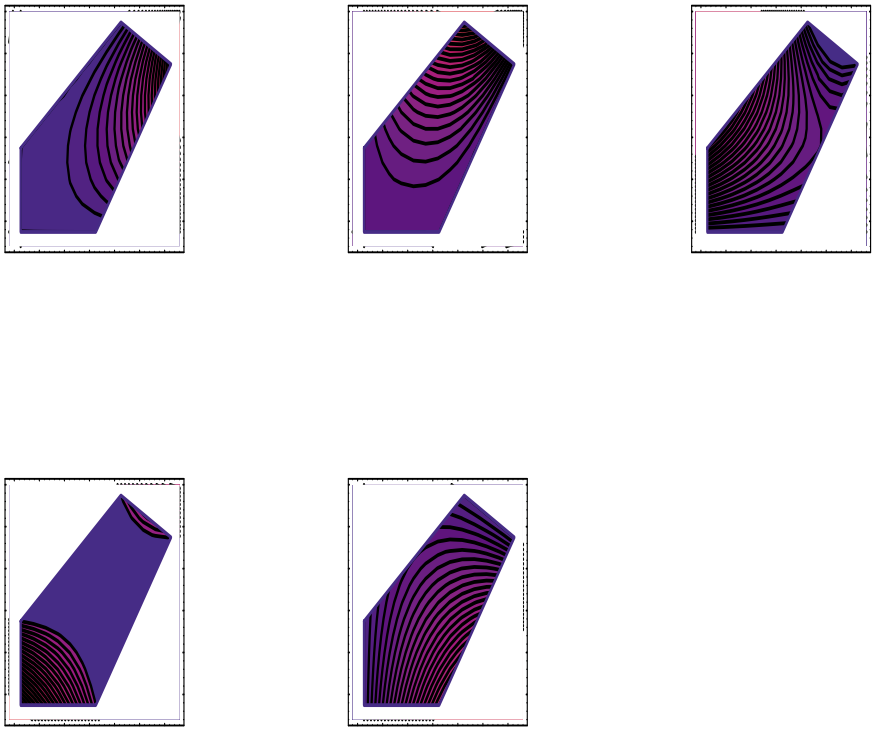

Figure 11: Contours plots for interpolants

The values of the interpolants at the nodes led to the identity matrix. This demonstartes that the boundary element formulation proposed here yields acceptable finite element rational polynomial interpolant shape functions.

On the element perimeter the distribution is not linear as demanded by the quadratic terms in equation (25). This in turn destroys the max/min characteristics of interpolation. The color plotting package reported that the blue color values overshot to 1.07317 for the given values of 1.0 at nodes 2 and 3. This error will be masked by interpolating linearly along the boundary within a (five to ten percent) band.

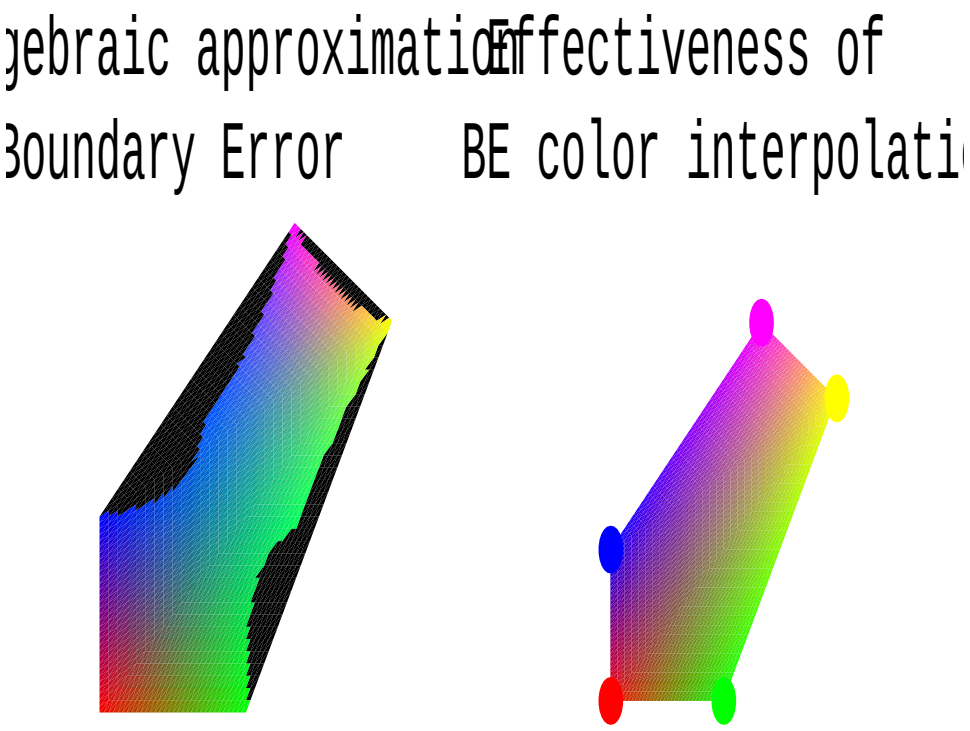

Figure 12: Comparison: BE *vis a vis* First Order Algebraic Approximation

4 Conclusions

Compact representation of computing information, i.e., compression of data without losing details, is an ongoing research in theoretical computer science since the inception of computers. This notion of program size complexity [3] along with the constructs of computational geometry [4] provide the algebraic foundation of computer graphics. Associated concepts of projective and perspective geometry can then faithfully define screen images from scientific experiments. For data visualization [8] and internet commerce, currently a substantial effort is devoted to develop closed form analytical (spline fitting) expressions to preserve image integrity. A numerical analog of such computer algebra programs [16] is explored here with any existing boundary element code.

Digital communication in the health care industry has been appreciably enhanced by the transmition of medical records on the internet and ability to receive live expert opinions on-line. (For example, the medical school of Columbia University has created a facility for medical informatics; Clinique Pasteur, Toulouse, France, is connected to a number of world class medical research institutions via satellite.) A number of diagnosis is routinely carried out by inspecting only time dependent shape changes of monochromatic graphics depicting radiological measurements. The kinematic consideration of continuum mechanics accurately quantifies such morphometry. Consequently, engineering analysis, which was formerly confined to the arena of conventional manufacturing and design, is attracting industrial applications in medical imaging.

Domains with an arbitrary boundary are represented as a collection of convex tessera, which make existing boundary element programs adequate for shape change calculation.

It is desirable to execute the Wachspress rational finite elements [15]based on exact interpolants on convex $n-$gons by executing a fast numerical code written in (Fortran or C++) a procedural language The influence functions are rational polynomials of order $n-2$ (numerator) by $n-3$ (denominator). Currently, the algebraic formulation with projective geometry constructs must be written in closed form (in a symbolic computer language) to preserve linearity on the edges. Inherent design of symbolic (higher level) languages, e.g., *Mathematica* and `Matlab`, slows down numerical simulations. Alternatively, an existing boundary element code written in Fortran or C++ (lower level) language can yield fast approximations.

An elliptic field equation is identified here for a boundary element scheme to approximate the Wachspress solution. The singularity, which does not disregard the overall aspect ratio of convex images, is derived from perspective geometry. The resulting departure from strict linearity on the boundary may generate slightly visible noise near the edges.

A color plot, which can simultaneously display more than one response on the same physical object, is indispensable in controlling and monitoring remote experiments. Tijani and Wacogne [9] pioneered *lab on a chip* to evaluate skin properties *in vivo.* A web based protocol can precisely apply the indentation force by interacting with color contours depicting instantaneous stresses, strains and displacement in a silicon beam with nonlinear force-deformation characteristics.

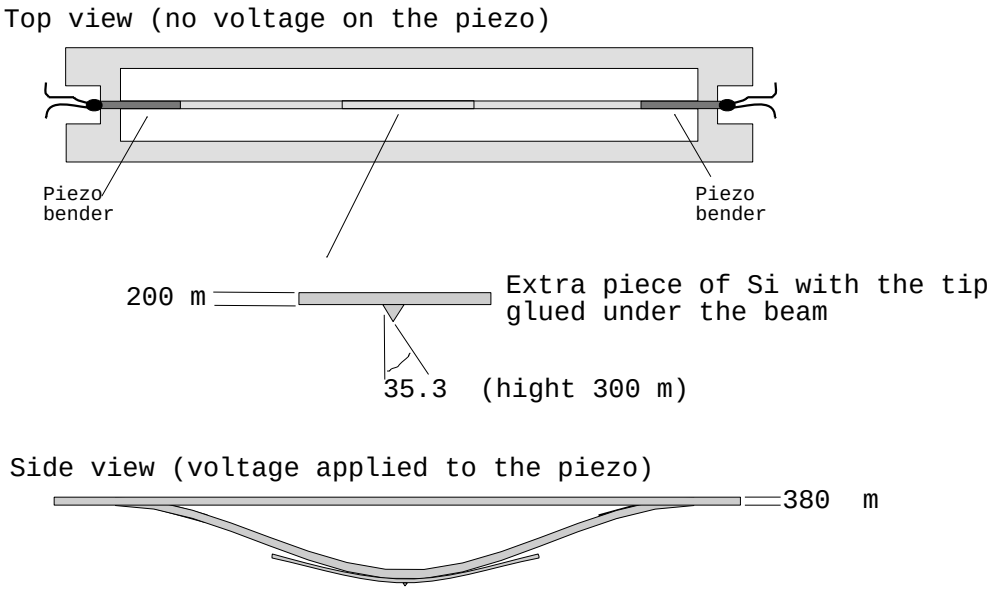

Figure 13: Tijani- Wacogne setup

For remote monitoring color contours of continuous stress or strain can be obtained from the proposed boundary element formulation. This facilitates online updating and control of such state-of-art experiments. Influence functions, which were developed for shape changes, are recommended to interpolate boundary colors, which represent different measurements at the piezioelectric probe points. This is a remarkable enhancement in rendering visual effects on a complex scientific figure by using a single boundary element code. The resulting nonlinear coloring on the edges does not seem to incur undesirable unevenness, which can be fixed by a linear masking at a small overhead. Of course, a boundary element code based on the (isotropic) Laplacian, can paint even disconnected (and concave) domains when a secondary anisotropic abberation in color perception is permitted.

Finally, on convex objects rational polynomial approximations of influence functions are developed circumventing the need of any boundary element code. Three interpolants out of n in a $n-$gon may violate the maximum/minimum principle, which is germain to solutions of elliptic partial differential equations. The scheme can be programmed in Java on the client side in live web implementation .

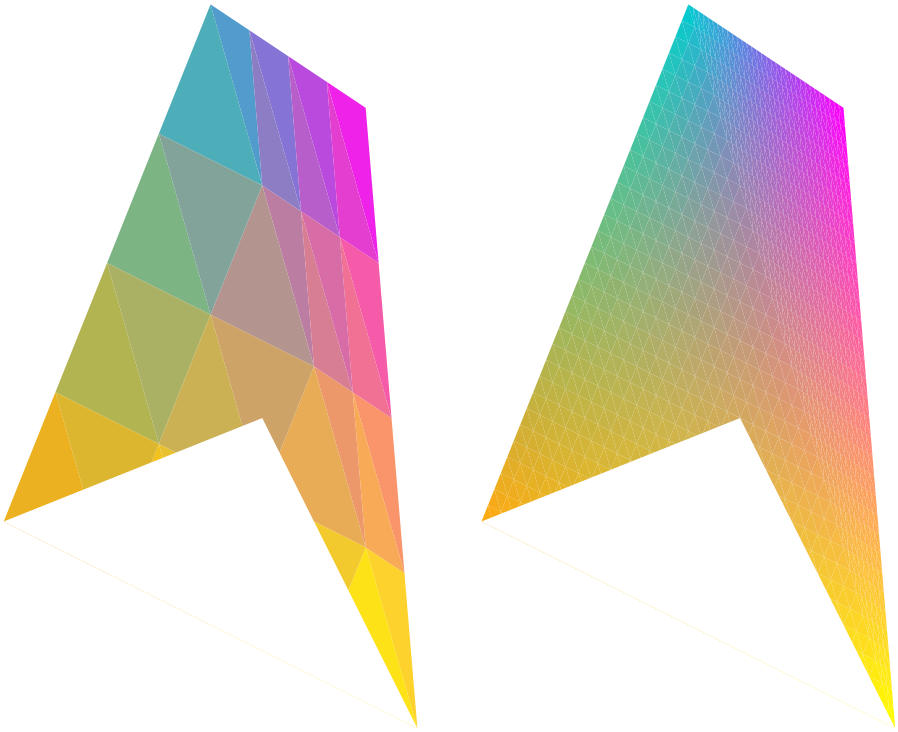

Figure 14: Triangulation vs BE color plotting of growth gradient

The boundary element coloring scheme proposed in this paper is compared in Figure 14 with the popular triangulation scheme to display a growth field related to the Treil's Maxillo-facial frame shown in Figure 5.

A computer algebra environment was essential for reducing lengthy closed form expressions leading to the anisotropic constant in the governing elliptic partial differential equation. As web browsers incorporate computer mathematics programs, such as the *webMATHEMATICA* `http://www.wolfram.com/products/webmathematica/`, highly accurate boundary elements with arbitrary singular kernels are anticipated to enrich the field of information technology where color changes in space and time occupy the center stage.

References

[1] BEASY. *BEASY User's manuals*. Computational Mechanics Publications, Ashurst, Southampton, UK, 2000.

[2] Steven Brooks and et. al. *Maya 2.5 Release Notes*. Alias Wavefront. Silicon Graphics Inc., 1999.

[3] Gregory J. Chaitin. *The Unknowable*. Springer Series in Discrete Mathematics and Theoretical Computer Science. Springer Verlag, 1999.

[4] Bernard Chazelle and et. al. Application challenges to computational geometry. *Computational Geometry Impact Task Force*, 2000.

[5] James D. Foley, Andries Van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics Principles and Practice*. Addison-Wesley, 2 edition, 1996.

[6] H. Hoppe. Progressive meshes. In *International Conference on Computer Graphics and Interactive Techniques*, volume 23, pages 99–108, New York, NY, 1996. SIGGRAPH, ACM.

[7] B. M. Irons and A. Razzaque. *Experience with the patch test for convergence of finite elements method*. Academic Press, New York, 1972.

[8] Shoichiro Nakamura. *Numerical Analysis and Graphics Visualization With Matlab*. Prentice Hall, 1995.

[9] A. G. Podoleanu, J. A. Rogers, Bruno Wacogne, S. Dunne, H. Porte, T. Gharbi, G. Dobre, and D.A. Jackson. Twords 3d oct imaging. In *Conference SPIE, Coherence Domain Optical Methods in Biomedical Science and Clinical Applications*, volume 3915, pages 46–54, San Jose, California, 2000. Summer Bioengineering Conference, Proceedings SPIE.

[10] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Graduate Textbook. Springer Verlag, 1985.

[11] David F. Rogers and J. Alan Adams. *Mathematical Elements for Computer Graphics*. McGraw Hill, 2nd edition, 1990.

[12] Mark Segal and Kurt Akeley. The design of the opengl graphics interface. Technical report, Silicon Graphics Inc., 1994.

[13] I. C. Taig. Structural analysis by the matrix displacement method. Report S017, English Electric Aviation Report, England, 1961.

[14] Jacques Treil and Gautam Dasgupta. Maxillo-facial model. *The Mathematica Journal*, 6 2001.

[15] E. L. Wachspress. *A rational finite element basis*. Academic Press, 1975.

[16] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, 2001.

Appendix-I

Determination of c_o in equation (12)

```
Start a session:
In[1]:= Clear[f, x, y, g, a, b, ell, t, n, p, q]

Define a boundary line segment with intercept form of straight lines,
a and b to be x and y intercepts, respectively:
In[2]:= ell[i_] := a[i] x + b[i] y + 1

Satisfy zero value along n boundary lines not cotaing the first node:
In[3]:=  n = 4; t[x, y] = Product[ell[i], {i, n}];
Here a hexagon (with n+2 sides) is demonstrated.

Define the first interpolant with singularity along the straight line joining
the first two EIPs:
In[4]:= f[x, y] = t[x, y]/(1 + p x + q y);

Write the field equation:
In[5]:= eq = D[f[x, y], {x, 2}] + Together[g[x, y]D[f[x, y], {y, 2}]] == 0;

Determine the unknown function g[x,y]:
In[6]:= sol = Solve[eq, g[x, y]] // Flatten;

First order approximation of  g[x,y] with centridal value:
In[7]:= g0 = Simplify[(g[x, y] /. sol) /. {x -> 0, y -> 0}]

Out[7]= -  (p^  2 + a[2] a[3] + a[2] a[4] + a[3] a[4] +
          a[1] (a[2] + a[3] + a[4])   - p (a[1]+ a[2]+ a[3]+ a[4]) ) /
(q^2 + b[2] b[3]+ b[2] b[4]+ b[3] b[4]+ b[1](b[2]+ b[3]+ b[4]) -
          q (b[1]+ b[2]+ b[3]+ b[4]))
Observe that p appears only in the numerator and q  only in the denominator:
In[8]:= CoefficientList[g0 // Numerator, p]
Out[8]={-a[2] a[3] - a[2] a[4] - a[3] a[4] - a[1] (a[2] + a[3] + a[4]),
                                  a[1] + a[2] + a[3] + a[4], -1}
In[9]:=CoefficientList[g0 // Denominator, q]
Out[9]={b[2] b[3] + b[2] b[4] + b[3] b[4] + b[1] (b[2] + b[3] + b[4]),
                                    -b[1] - b[2] - b[3] - b[4], 1}
```

Appendix-II

Shape Relations

```
Clear[allShapes, shapeRelation]

shapeRelation2D::usage = "shapeRelation[f,s1,{x,y},nodes]generates {mat,vec}
for determining the shape functions as LinearSolve[mat, vec] with the initial
shape function s1 in {x,y} coordinate system associated with  nodes by
exactly interpolating the elements of f in the coordinate system (x,y)."

allShapes::usage = "allShapes[f,s1,{x,y},nodes] yields the complete set of
shape functions  with the initial shape function s1 in {x,y} coordinate
system associated with  nodes by exactly interpolating the elements of f."

Clear[allShapes, shapeRelation]

shapeRelation[f_, s1_, z_, nodes_]:= {Table[f /. Thread[z -> nodes[[i +1]]]
                   , {i, Length[f]}] // Thread,
                                  f - s1(f /. Thread[z -> First[nodes]])}

allShapes[f_List, s1_, z_, nodes_] := Prepend[
      LinearSolve @@ (shapeRelation[f, s1, z, nodes]), s1] // Factor
```