

## ALGORITHM DEVELOPMENT AND IMPLEMENTATION FOR NUMERICAL DETECTION OF HOT SPOTS IN CASTING GEOMETRY

Feroz Shah\*, Mohammad A. Irfan\*\*, Rizwan Gul\*, Abdul Shakoor\*

### ABSTRACT

*Numerical prediction of hot spots inside solidifying castings is presented. Casting geometry is decomposed into rectangular sections. Arbitrary points are chosen in these sections from where vectors are generated which interact with surrounding edges, corners, curved surfaces, holes etc. of the geometry. These vectors develop a network of triangular segments, which define the boundaries of rectangular sections. MATLAB functions defined for rectangular sections decide the relative rectangular section for an arbitrary point. During implementation in MATLAB, 360 vectors are generated from an arbitrary point, through 1° to 360°, in a closed circle. Magnitude and direction of averaged values for all the vectors are computed for each iteration. The new point is moved relative to the previous point depending upon magnitude and direction of the average value of all the vectors. After few iterations the point converges inside the geometry which is recognized as the hot spot. This technique is sensitive to the influence of corners, cores, curved profile, fins, chills etc. Hot spots for several sections, i.e., L, T, U, Stepped geometry and an engine saddle with and without holes were located. This technique shows good agreement to location of shrinkage cavity or porosity observed in cast sections.*

**KEYWORDS:** Hot spot, porosity, numerical technique.

### 1. INTRODUCTION

Casting is one of the most economical manufacturing processes used for shaping metals and alloys. It is a complicated process which involves significant metallurgical and mechanical aspects. A well known defect related with cast products is porosity, which can adversely affect mechanical performance as well as the aesthetics of the cast product. Depending on the application, defects associated with porosity can be costly as considerable value can be added to the casting during the various manufacturing stages prior to detection. Casting of metals is a complicated process and is governed by number of variables, viz. casting alloy, mould material, casting limits, mould height, size and location of feeders, runners and ingates, degree of superheat before casting, casting pressure etc. Different materials show their own unique solidification patterns depending upon. Eutectics and pure metals solidify at constant temperatures. Binary and ternary alloys solidify over a range of temperatures. Simulation of solidification process is used to identify the defect inside casting geometry<sup>1</sup>. Several factors render difficulties in analyzing solidification of casting alloys inside a mould, viz. transient heat transfer involving latent heat generation due to unstable liquid-solid boundary, complex casting geometry, non-uniform thickness of the castings, metal-mould interface, local re-melting of the

alloys, fins, holes, chills, feeder insulation, exothermic cores etc<sup>2</sup>. Detection of hot spot before actual casting of metals inside a casting geometry is very beneficial and can be used to take preventive action to eliminate hot spot in sections of mechanical importance. Hot spot is a potential spot for shrinkage and accumulation of gas porosities.

### 2. REVIEW OF TECHNIQUES USED FOR PREDICTION OF HOT SPOT

Simulation of casting solidification has been generally used to predict solidification related defects like shrinkage cavity, porosity, microstructure, residual stresses etc. During solidification, liquid-solid interface progressively shift from the mold boundary till it reaches to certain point/points in the mold-cavity, where interface vanishes, is generally called as hot spot/spots. Generally, there are two different approaches adopted to simulate solidification of castings viz.

- A. Physics based.
- B. Geometry based simulations.

Both of the approaches have been explored by many researchers for casting simulation. Physics based

\* Department of Mechanical Engineering, University of Engineering & Technology, Peshawar, Pakistan

\*\* Department of Mechanical Engineering, Qassim University, Saudia Arabia

approaches are inclusive but involve more computation time, and have greater hardware requirements. Geometry based techniques are basic and require less computational time but lack in taking into account the physics of thermal boundary limits.

### A. Physics Based Methods

The following physics based numerical techniques for casting solidification simulations are available in market<sup>2</sup>:

- Finite difference method (FDM)
- Finite element method (FEM)
- Boundary element method (BEM)

Finite difference method (FDM) is one of the oldest numerical techniques used to generate the solidification simulation<sup>4</sup>. The FD method provides easy mesh generation and less storage for defining the geometry. This enables it to simplify the implementation of the numerical algorithms<sup>5</sup>. During solidification the resistance due to air gap between casting and mould wall is taken into account and is solved using one of the above mentioned techniques<sup>6-8</sup>. Different modes of heat transfer viz. conduction, convection and radiation can be modeled by writing heat transfer equations for nodal elements in FEM. Velocity of solid-liquid interface, time- temperature distribution during solidification can be computed by discretizing the geometry and development of stiffness matrices<sup>9,10</sup>. Boundary element method is considered to yield more accurate result outputs and is regarded as an advanced technique by engineering scientists and technologists<sup>11,12</sup>. Several investigations have been carried out for locating hot spots using Finite difference method and FEA through heat transfer computation<sup>13,14</sup>. FEA requires extensive knowledge of thermo-physical properties of cast and mould alloys at various temperature ranges. Component geometry and its relevant boundary limit needs to be clearly defined for FEA simulations.

Several programs for solidification simulation and feeder design are available, with one or other limitations. Some of the casting simulation softwares that use one of these numerical techniques are AFSOLID,

AUTOCAST, MEGMASOFT, NOVASOLID, ProCAST, SIMTEC, FLOW-3D, etc<sup>15,16</sup>.

### B. Geometry Based Methods

Chvorinov<sup>17</sup>, modulus computation method is related to specific shapes, metal or alloys and mould. It relies on user interaction for decomposition of the component section analysis. Heins modulus method was used to simulate solidification front and was demonstrated on shapes such as plate-feeder or gear wheel sections. Kotchi<sup>18</sup> developed a method for simulating the thickness of metal frozen based on volume/surface area contour lines. Ravi<sup>2</sup> has developed geometry based vector modulus technique for the detection of hot spot. In this method the geometry of castings is divided into a numbers of wedged each subtending same specified angle with its apex at the chosen point within the section. The averaged heat flux corresponding to the wedge is given by the ratio of enthalpy of freezing to the heat transfer. Resultant for Thermal vectors for each wedge is calculated. Hot spot will lie where the resultant for the point is zero. The current investigation has lead to the development of geometry based numerical simulation technique for prediction of hot spot using MATLAB coding. This program is free of graphical softwares but instead numerically generates an image of casting geometry and plot points on that image sequentially that leads to a point inside a casting geometry known as hot spot.

## 3. GEOMETRY BASED NUMERICAL TECHNIQUE

The geometry based numerical technique presented here for the prediction of hot spot, assumes that heat transfer from the casting section depend on the shape of the casting geometry and a uniform thickness in third dimension. Casting geometry is divided into number of rectangular sections depending upon the complexity and curves/ holes inside the geometry. In Figure1, L and T shaped geometries have been divided into respective rectangular sections. L and T sections have been divided into 3 and 4 rectangular sections respectively.

An arbitrary point inside each rectangular section is marked and vectors are generated from the point to interact with surfaces of the geometry surrounding the point. These vectors will form triangular segments of

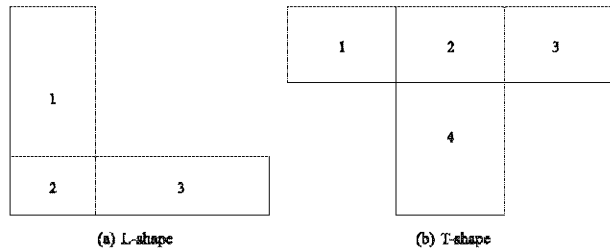


Figure 1: Division of geometries into rectangular sections.

different sizes depending upon the interaction of their bounded vectors with surrounding geometrical features, viz. corners, under cuts, nodes of curved surfaces/holes, Figure 2.

The triangular segments will cover all the geometrical features relating to that particular rectangular section that may include holes, corners and curved features.

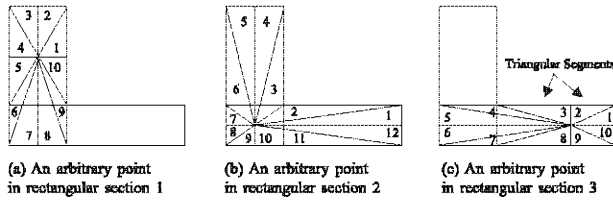


Figure 2: Vectors generation from an arbitrary point in rectangular sections.

Therefore, the numbers of triangular segments depend upon the included features of that particular rectangular section, Figure 3. Different triangular segments will subtend different angles at common vertex (arbitrary point). The angle subtended by each triangular segment is calculated through trigonometric ratios.

As can be seen in Figure 3, the angle subtended by each triangular segments viz. 7, 8, 9 is calculated through

$$x = y * \cotan(\theta) \quad (1)$$

$$y = x * \tan(\theta) \quad (2)$$

$$\theta = \tan^{-1} (y/x) \quad (3)$$

trigonometric ratios. The ending angle of preceding triangular segment becomes the starting angle for the subsequent segment and in this way all the triangular segments covers 360° collectively.

After defining the boundary limits for each triangular

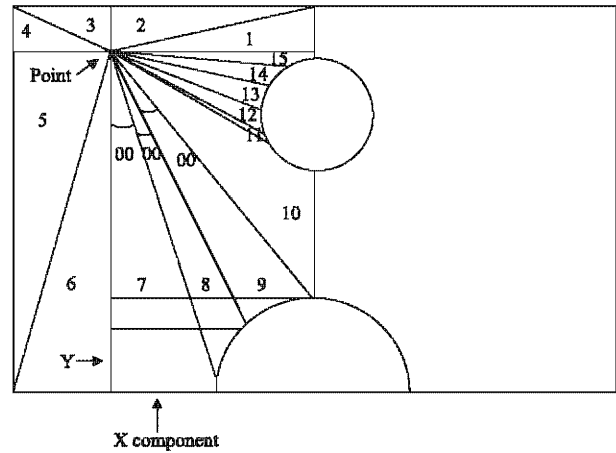


Figure 3: Calculation of boundary Limits for triangular segments

segment, the algorithm is implemented in MATLAB as shown in Figure 4.

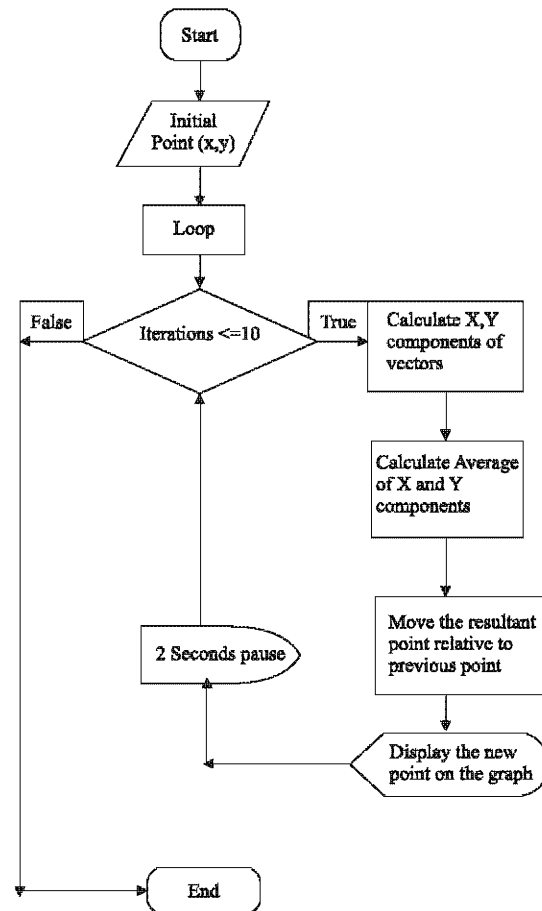


Figure 4: Flow diagram for the algorithm developed for prediction of hot spot.

In MATLAB an arbitrary point is chosen in any of the rectangular section. Specific MATLAB functions defined for rectangular sections automatically recognize the location of the arbitrary point in the geometry. Another set of vectors generated from this point through  $360^\circ$  at an increment of  $1^\circ$ , so a total of 360 vectors for each point, shown in Figure 5. One degree increment is used to minimize the spacing between the vectors for better accuracy. Magnitude and direction of the resultant of all 360 vectors is calculated and the arbitrary point is moved relative to previous position in the direction of the resultant. After several iterations the relative difference between the points decreases and consequently converges to a point known as hot spot. Holes, undercuts and curved boundaries are automatically handled by the lengths of the corresponding vectors.

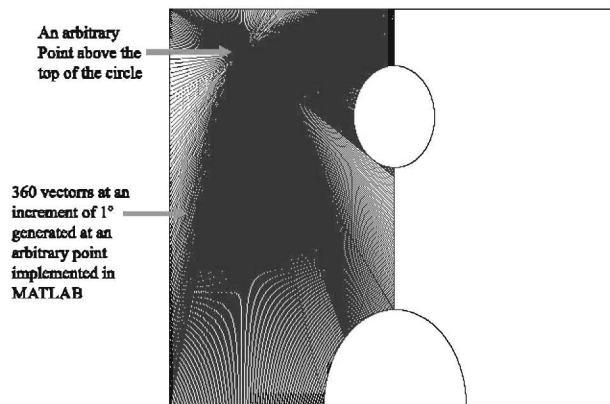


Figure 5: Generation of 360 vectors at an arbitrary point.

## 4. RESULTS AND DISCUSSIONS

### A. Analytical Development of Algorithm

Figure 6 shows the section of an engine block. The section is taken from a place near the crankshaft bearing<sup>19</sup>. The geometry is referred to a “saddle” in this paper. As a particular case, detail explanation for locating hot spot for saddle with hole given in Figure 14 is presented below.

1. **Symmetrical Division of the Geometry:** The geometry of the saddle with a hole is first divided symmetrically into two halves (Figure 7).

2. **Marking of Quadrant for Division of Circular and Semicircular sections in the Geometry:** Quadrants for circular hole and semicircular saddle are marked so that

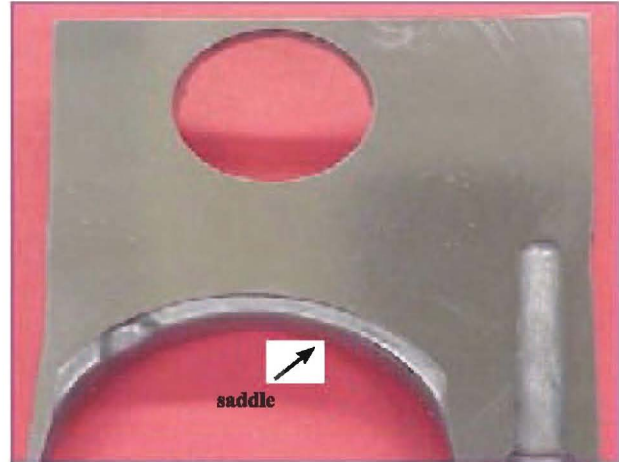


Figure 6: Saddle section of the engine block.

the effect of curviness of the circular sections is taken care of in calculating length of the vectors. The numbers of division on the periphery of curved surfaces linked to the effect of the curviness of the surfaces inside a geometry. The length of the vectors controlled the curviness of the curved surfaces as well as undercuts, key ways etc. generated from an arbitrary point. The circle in the geometry is divided into six equal parts subtending an angle of  $30^\circ$  each. The semi circular arc is divided into three equal parts each extending an angle of  $45^\circ$  (Figure 7).

3. **Division of Geometry into Rectangular sections:** The left and right halves have been divided into four rectangular sections according to the complexity of the chosen geometry (Figure 7).

4. **Analytical Algorithm for Left Piece:** The symmetrical left half of the geometry has been further divided into

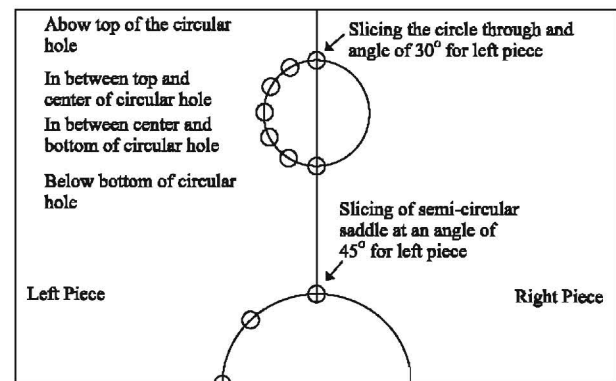


Figure 7: Division of the geometry for analytical algorithm.

four horizontal segments. An arbitrary point is taken in each of the segments namely

1. Above the top of the circle.
2. In between the top of the circle and center.
3. Between center and bottom of the circle.
4. Below bottom of the circle.

Vectors are generated from the point chosen that touches the boundaries of the geometry surrounding the point.

5. Point above the top of the circle for left piece: An arbitrary point above the top of the circle is chosen from where vectors are generated to strike the surrounding boundaries, viz. edges, corners, curved surfaces of the geometry. The angle between two vectors is calculated through trigonometric ratios that will serve as boundary limit for the each triangular section between the vectors (Figure 8).

6. Point in between the top and center of the circle: Same procedure is applied for this point but now it will refine the previous calculations because some of the corners or curves may have missed out due to orientation of the previous point, it serves like moving a camera around the object to capture all the details (Figure 9).

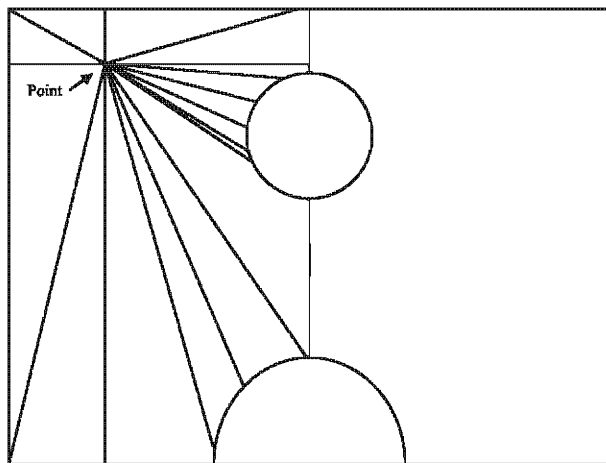


Figure 8: An arbitrary point in the geometry above the top of the circle.

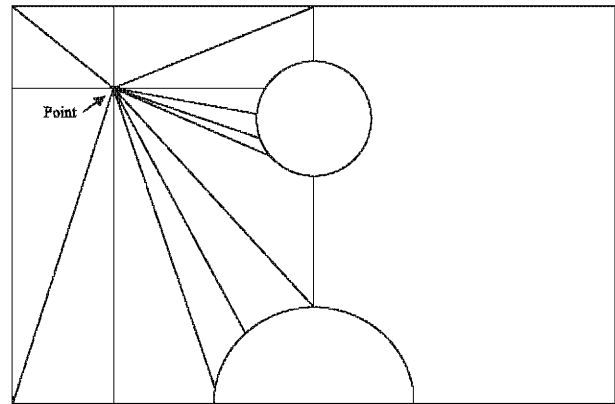


Figure 9: An arbitrary point in between top of the circle and center of the circle.

7. Point in between center and bottom of the circle: An arbitrary point in between the center and bottom of the circle is marked and calculation for the angles between the vectors is calculated, thus again boundary limit i.e. angle subtended by each triangular section is obtained (Figure 10).

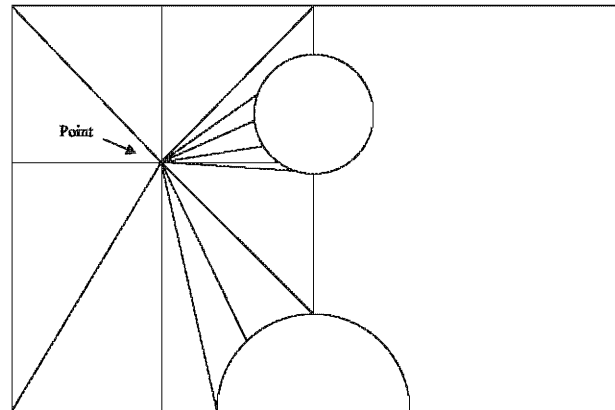


Figure 10: An arbitrary point in between center and bottom of the circle.

8. Point below the bottom of the circle: An arbitrary point below the bottom of the circle is marked and vectors are generated from it. This point will cover the bottom face of the circle and will refine the missing details from below (Figure 11).

9. Analytical algorithm for right piece: As was done for the left piece an arbitrary point for the right piece has been evaluated and boundary limit for each rectangular

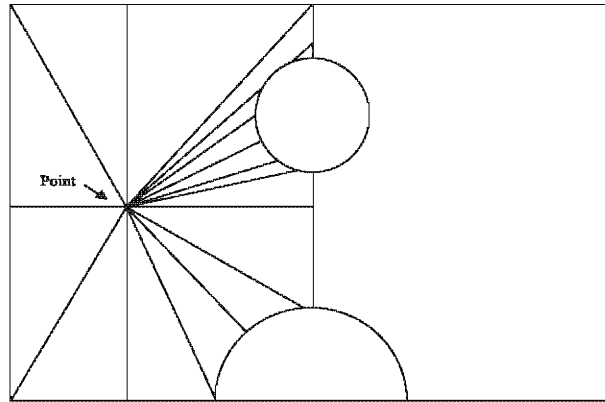


Figure 11: An arbitrary point below the bottom of circle.

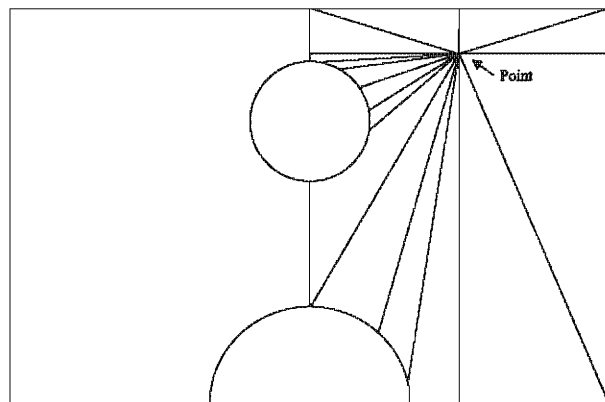


Figure 12: An arbitrary point above the top of circle for right piece.

section between two vectors were calculated using trigonometric ratios (Figure 12).

### B. Implementation of the Developed Algorithm in Matlab

Since the geometry has been divided into several rectangular sections during analytical algorithm development, separate MATLAB function is defined for each rectangular section. Depending upon the location of the arbitrary point, MATLAB will first decide the particular rectangular section containing the point. After locating the point, 360 vectors are generated at the point with an increment of  $1^\circ$ , through  $360^\circ$  in a close circle. Vectors are resolved into rectangular components and their averages are calculated. After first iteration we get a new point (X,Y) where X is the average of horizontal components and Y is the average of vertical components.

$$X = \sum_{i=1}^{360} \left( \frac{X_i}{360} \right) \quad (4)$$

$$Y = \sum_{i=1}^{360} \left( \frac{Y_i}{360} \right) \quad (5)$$

The function then returns calculated values for the point to the main function. The main function moves the new point relative to the previous point depending upon the magnitude and direction of the new point in X and Y directions. The main function then once again decides the location of the newly moved point and calls the specific function relating that section. 360 vectors are again generated from this point and their components are resolved and calculated. The process continues till the difference between the consecutive points minimizes to zero and thus a point of convergence is obtained. According to developed algorithm this point of convergence is the desired hot spot/spots (Figure 13).

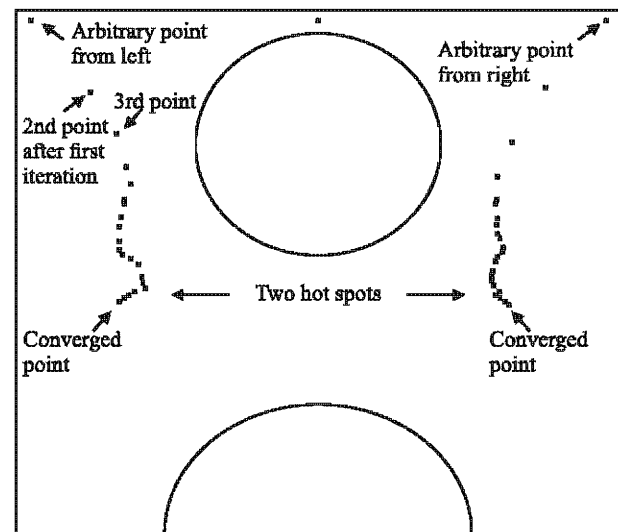


Figure 13: Two hot spots in saddle with a hole through MATLAB simulation.

This geometry based numerical technique depends on geometry of the castings sections and is independent on thermo- physical properties of the metal and mould materials. It is a sensitive and versatile method and requires less processing time and memory of the computer as compared to physics based numerical techniques. This technique can be use to rectify runners and riser/ feeder locations at design stages. Hot spots for several shapes have been programmed and executed through this algorithm, shown in Figure 14.

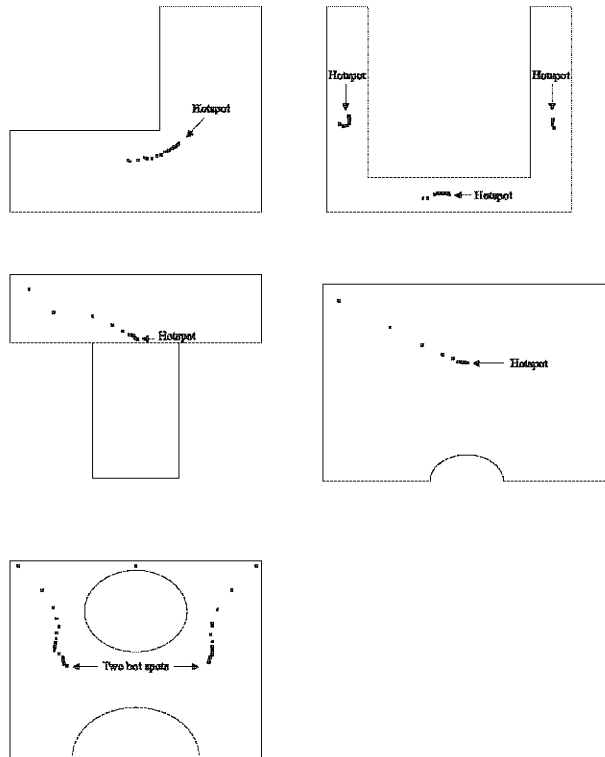


Figure 14: Hot spots for L, saddle with hole, Saddle without hole, T and U shape geometries.

The results are in strong agreements with that of the published simulated and experimental work<sup>2,19,20</sup>.

## ACKNOWLEDGMENT

Higher Education Commission (HEC) of Pakistan

## REFERENCES

1. Vijayaram T.R., Sualiman S., Hamouda A., and M. Ahmad, (2006) "Numerical simulation of casting solidification in permanent metallic moulds journal of materials processing technology," *Journal of Materials Processing Technology*, vol. 178, pp. 29–33.
2. Ravi B. and Srinivasan M. N., (1990) "Hot spots in casting: Computer aided location and experimental validation," *AFS Trans* 98, pp. 353–357.
3. Vijayaram T.R., (2008) "Metal casting solidification simulation technology," *Journal of Advanced Manufacturing Technology*, vol. 2, no. 2, pp. 19–28.
4. Chan I. and Stefanescu C. D., (1984) "Computer-aided differential thermal analysis of spheroidal and compacted graphite cast iron," *Transactions of American Foundrymen's Society*, vol. 92, pp. 947–96.
5. Kermanpur A., Mahmoudi S., Hajipour A., (2008) "Numerical simulation of metal flow and solidification in the multi-cavity casting moulds of automotive components," *Journal of materials processing technology*, vol. 206, no. 1-3, pp. 62–68.
6. Venkatesan A., Gopinath V.M., Rajadurai A., (2005) "Simulation of solidification and its grain structure prediction using fem," *Journal of Materials Processing Technology*, vol. 168, pp. 10–15.
7. Hou S., Zhao W., Ren F., (2002) "Progress in micro-modeling of the casting solidification process," *Journal of Materials Processing Technology*, vol. 123, no. 3, pp. 361–370.
8. Ho-Le K., (1988) "Finite element mesh generation methods a review and classification," *Computer Aided Design*, vol. 20, pp. 27–38.
9. Ravindran K., Lewis R., (1998) "Finite element modeling of solidification effects in mould filling," *Finite Element in Analysis and Design*, vol. 31, no. 2, pp. 99–116.
10. Oldfield W., (1966) "A quantitative approach to casting solidification: freezing of cast iron," *Transactions American Society of Metals*, vol. 59, pp. 945–959.
11. Pehlke R.D., Jeyarajan A., Wada H., (1982) "Summary of thermal properties for cast alloys and mold materials," *The University of Michigan, Ann Arbor*.
12. Dantzig J., Wiese J., Lu S., (1985) "Modeling of heat flow in sand castings part1 and part2," *Metallurgical Transaction*, vol. 16, no. B, pp. 195–209.
13. Berry J. T., Pehlke R. D., (1988) "Modeling of solidification heat transfer," *ASM Metals Handbook*, vol. 15., no. 9, pp. 858–866.