G

APPENDIX G: STEADY FLOW RESULTS OF CHOSEN IMPACT PAD DESIGN

G-1 INTRODUCTION

This appendix gives the detail flow results of the optimised impact pad obtained from case study 3. The appendix shows the temperature, flow distribution and turbulence levels in the tundish.

G-2 TEMPERATURE CONTOURS

The temperature contours in the tundish are first shown on the walls (Figure G-1 and Figure G-2) and then on vertical planes throughout the tundish width (Figure G-3 through Figure G-7). In Figure G-8, the temperature contours are shown on the slag layer (note that this figure has a reduced scale compared to the other temperature plots, all of which have the same scale). Starting with this last figure, the contours can be seen to be remarkably symmetrical on the slag surface even though the shroud is offset. At the bottom, especially inside the impact pad (Figure G-2), the side wing of the pad nearest the shroud position is notably hotter than the other wing. From the vertical planes (Figure G-3 through Figure G-7), it can be seen that a cold dead zone extends across the whole tundish width in the wake of the impact pad.

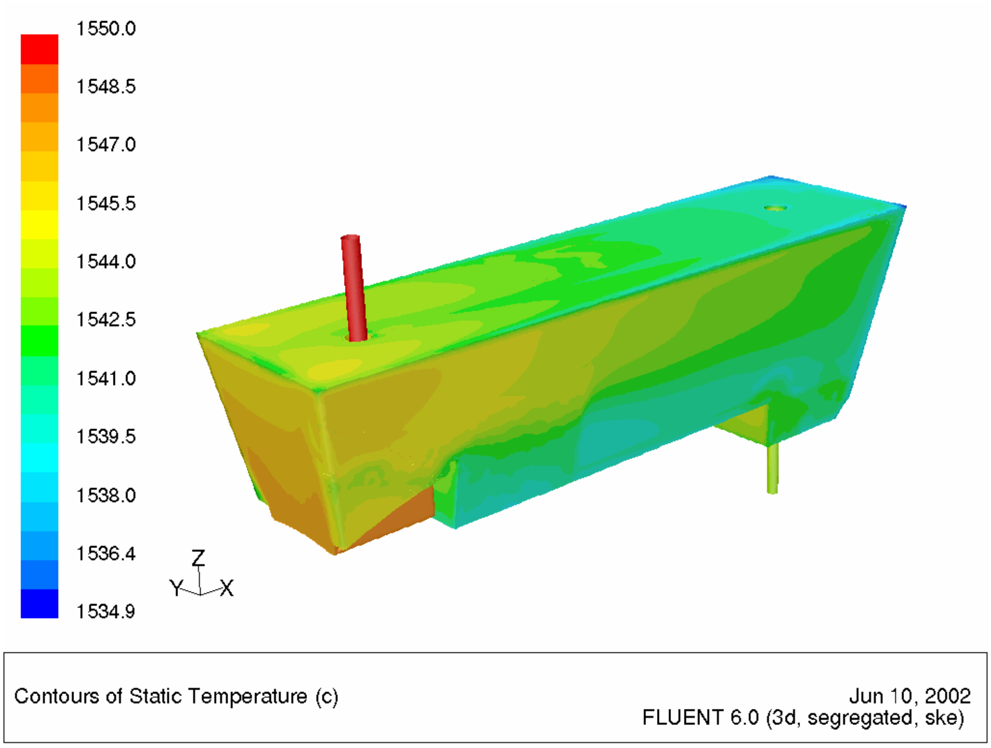

Figure G-1: Temperature contours on lining

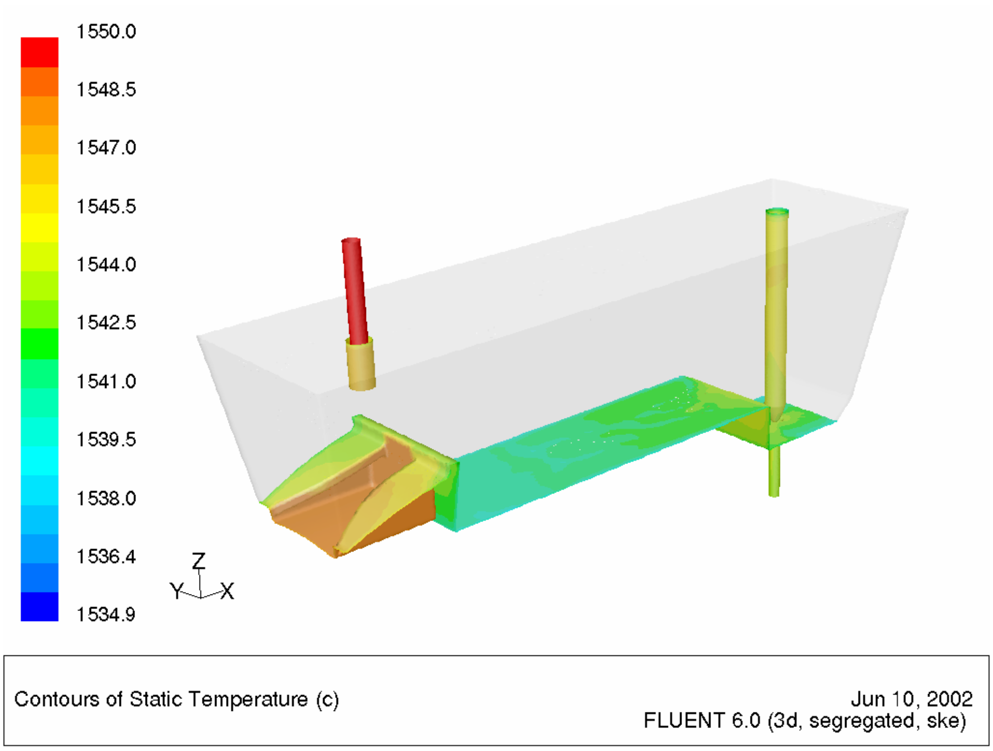

Figure G-2: Temperature contours on bottom wall and impact pad

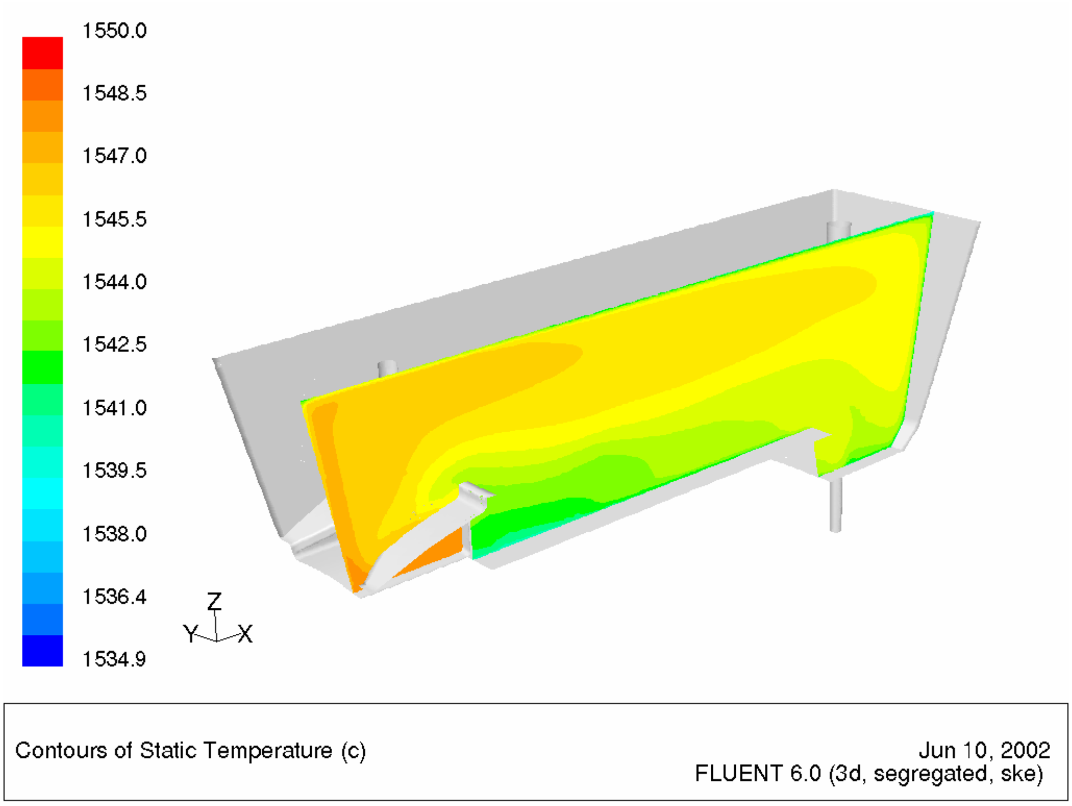

Figure G-3: Temperature contours on vertical plane – (y=0.2m)

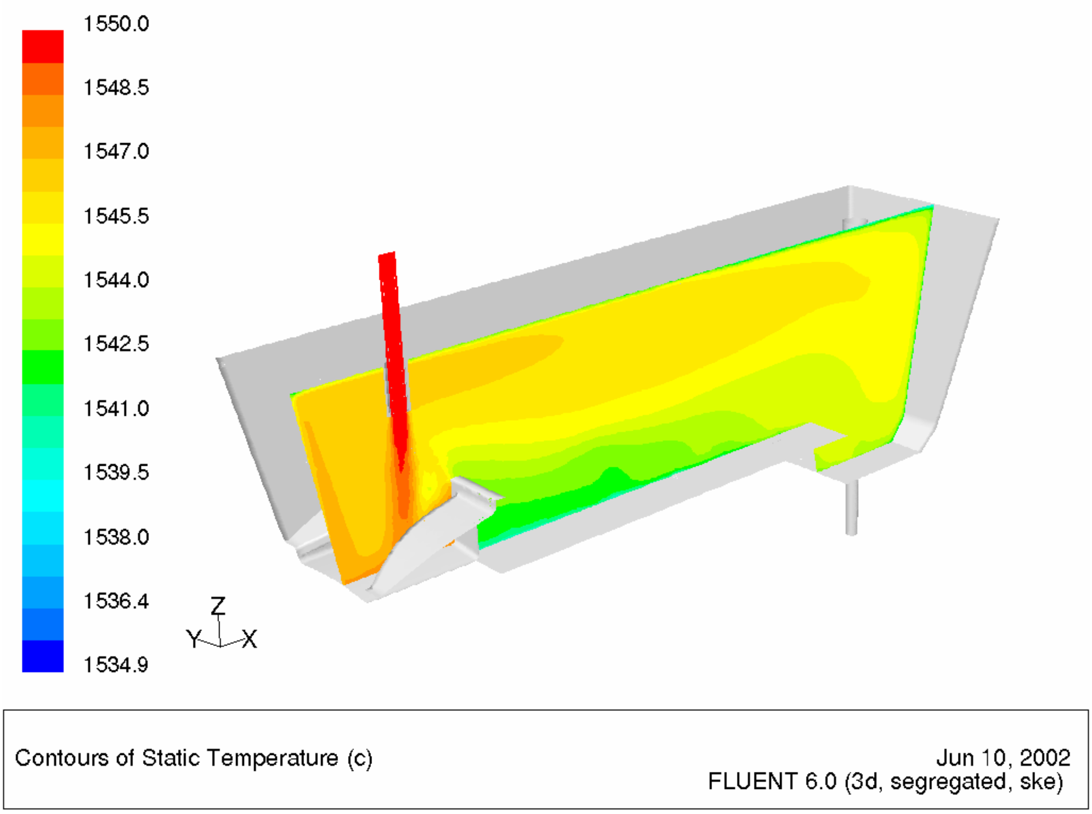

Figure G-4: Temperature contours on vertical plane – Centre shroud (y=0.32m)

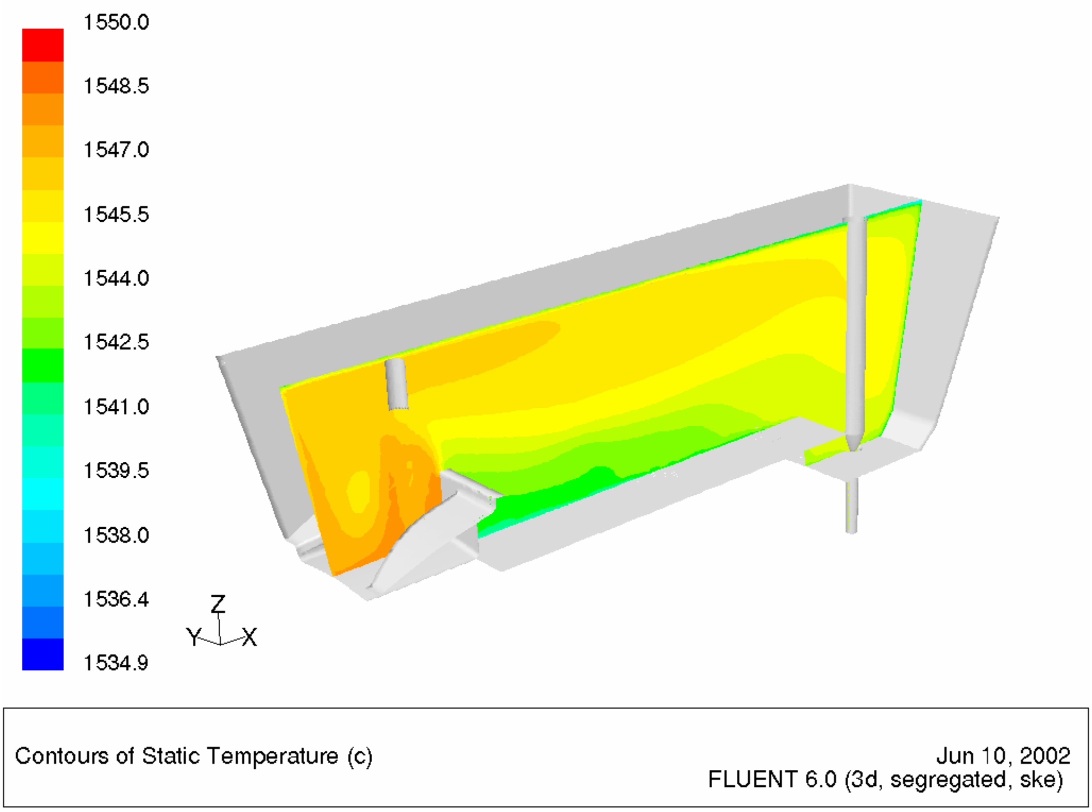

Figure G-5: Temperature contours on vertical plane – Centre SEN (y=0.4m)

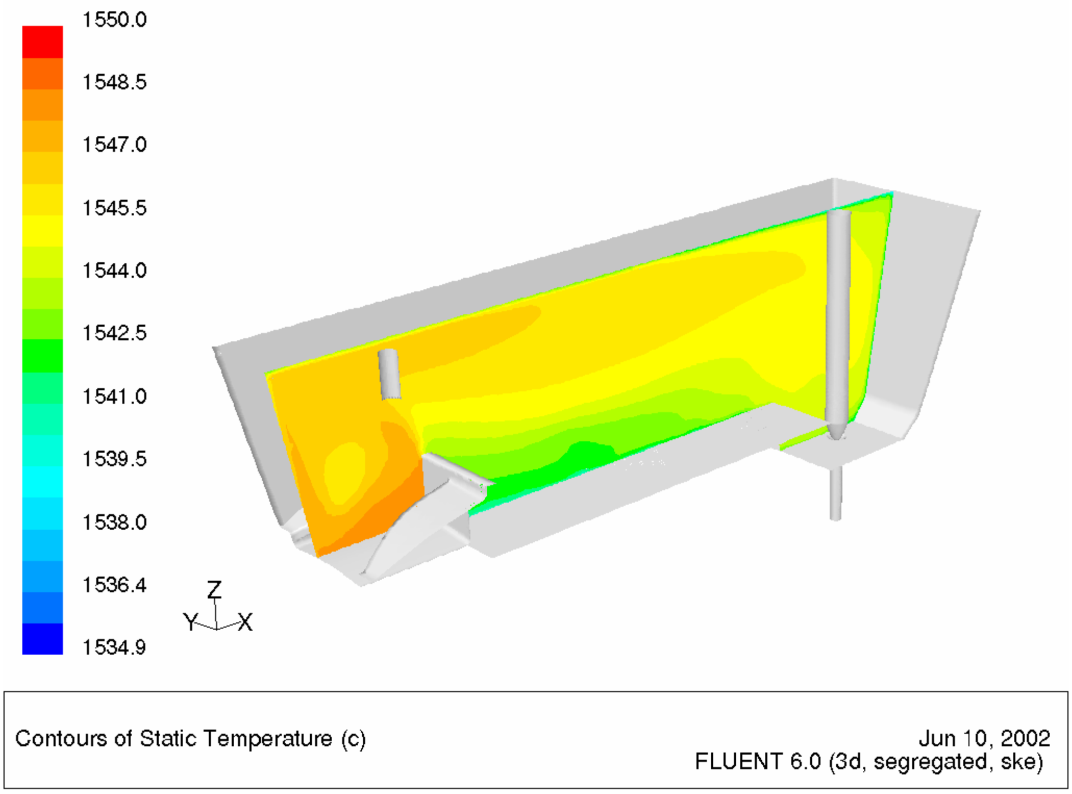

Figure G-6: Temperature contours on vertical plane – (y=0.48m)

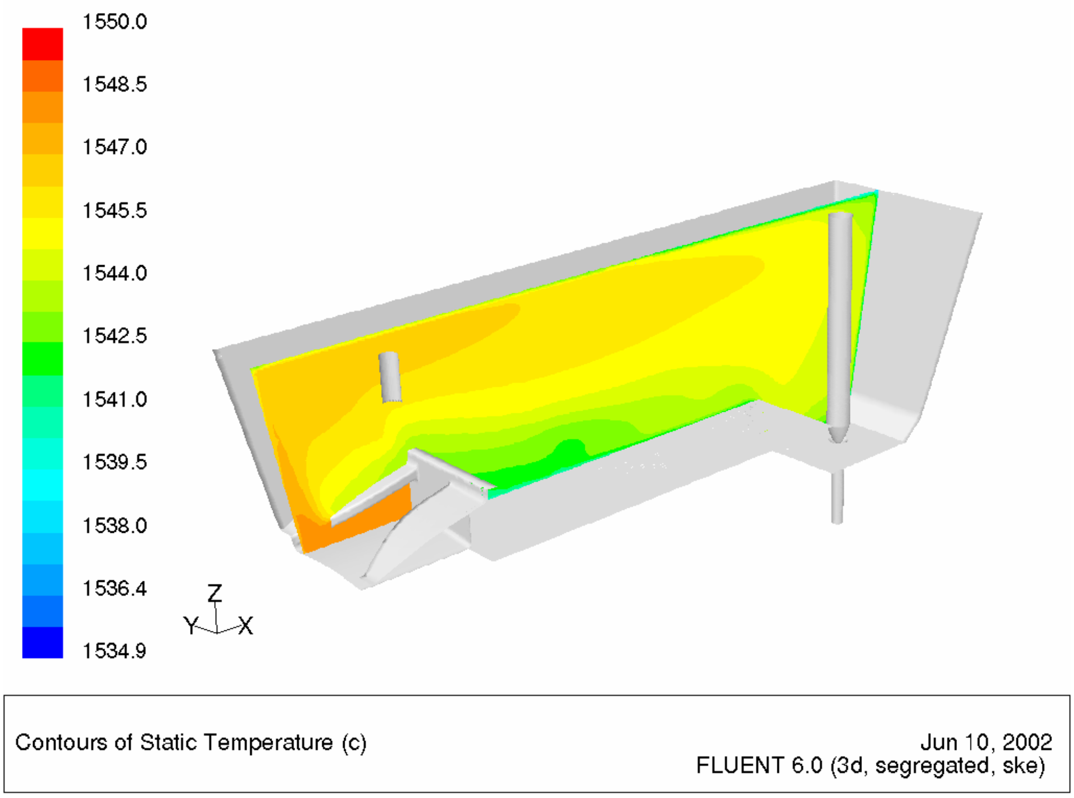

Figure G-7: Temperature contours on vertical plane – (y=0.6m)

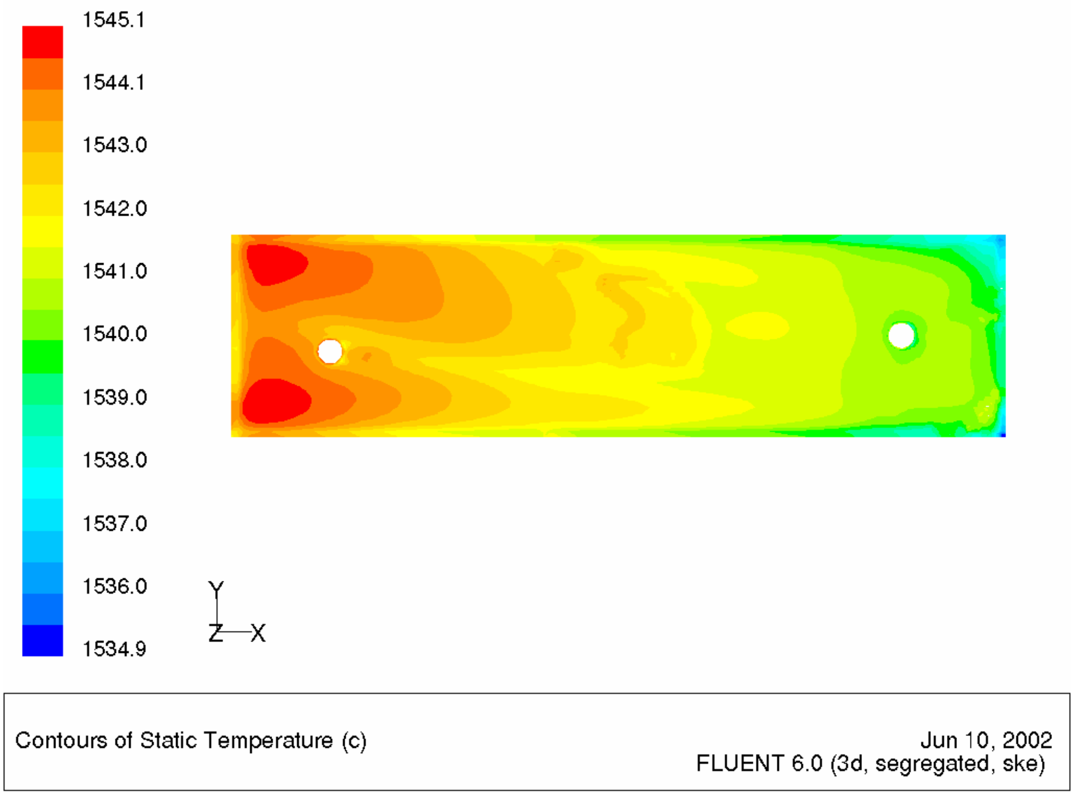

Figure G-8: Temperature contours on slag layer

G-3 PATH LINES AND CONTOURS OF VELOCITY MAGNITUDE

The path lines and contours of velocity magnitude in the tundish on vertical and horizontal planes are shown in Figure G-9 and Figure 10 through Figure G-20 respectively. All contour plots are clipped at 0.1m.s^{-1}. Note how the inlet jet is split into two. The one section continues up the back wall and extends along nearly the whole of the length of the tundish just below the slag layer. The other part of the jet forms a 'tongue' that is located at a median height in the tundish. This tongue is fairly two-dimensional in that it extends across the full width of the tundish. There is some evidence of three-dimensional flow due to the off-set shroud, but the impact pad does manage to limit the three-dimensionality of the flow.

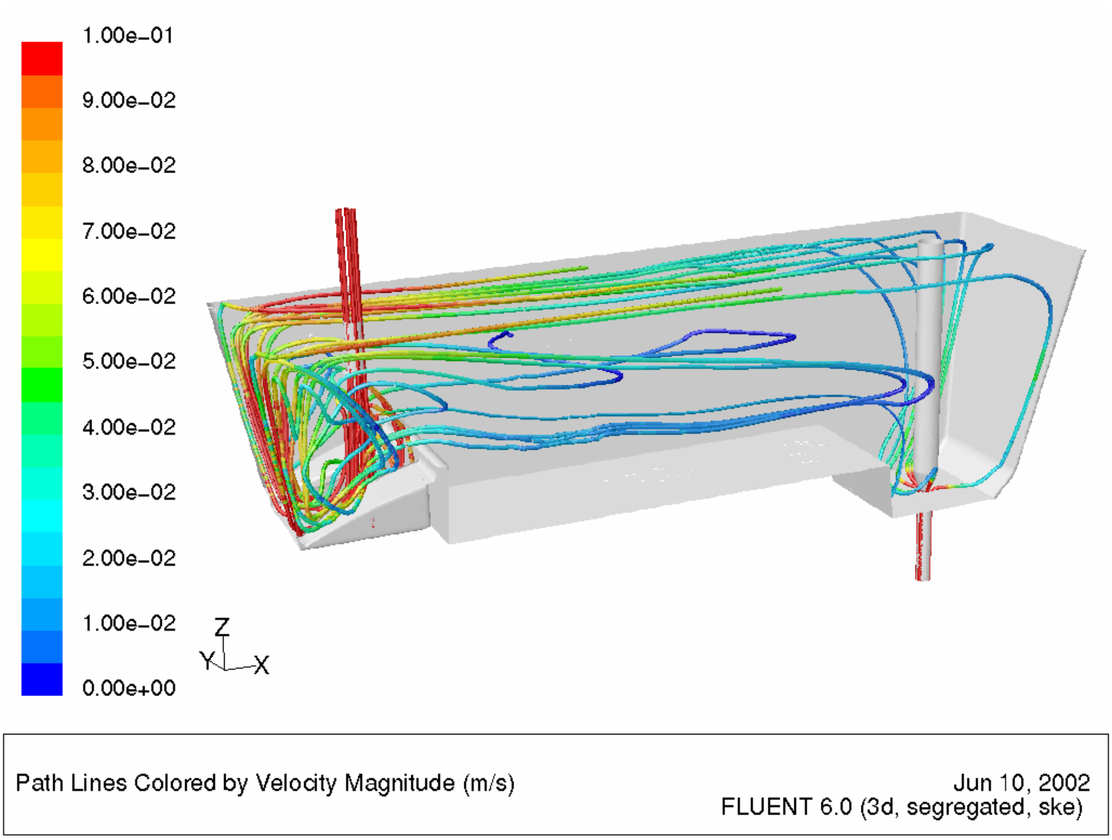

Figure G-9: Path lines in tundish coloured by velocity magnitude

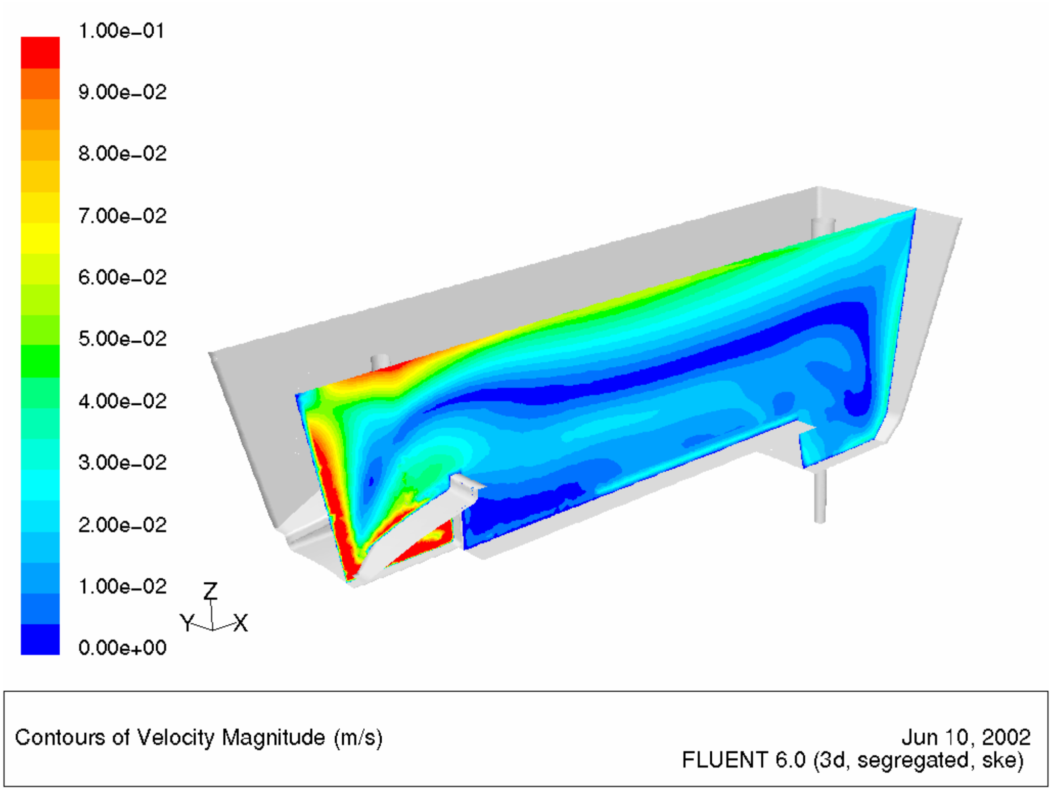

Figure 10: Velocity magnitude contours on vertical plane – (y=0.2m)

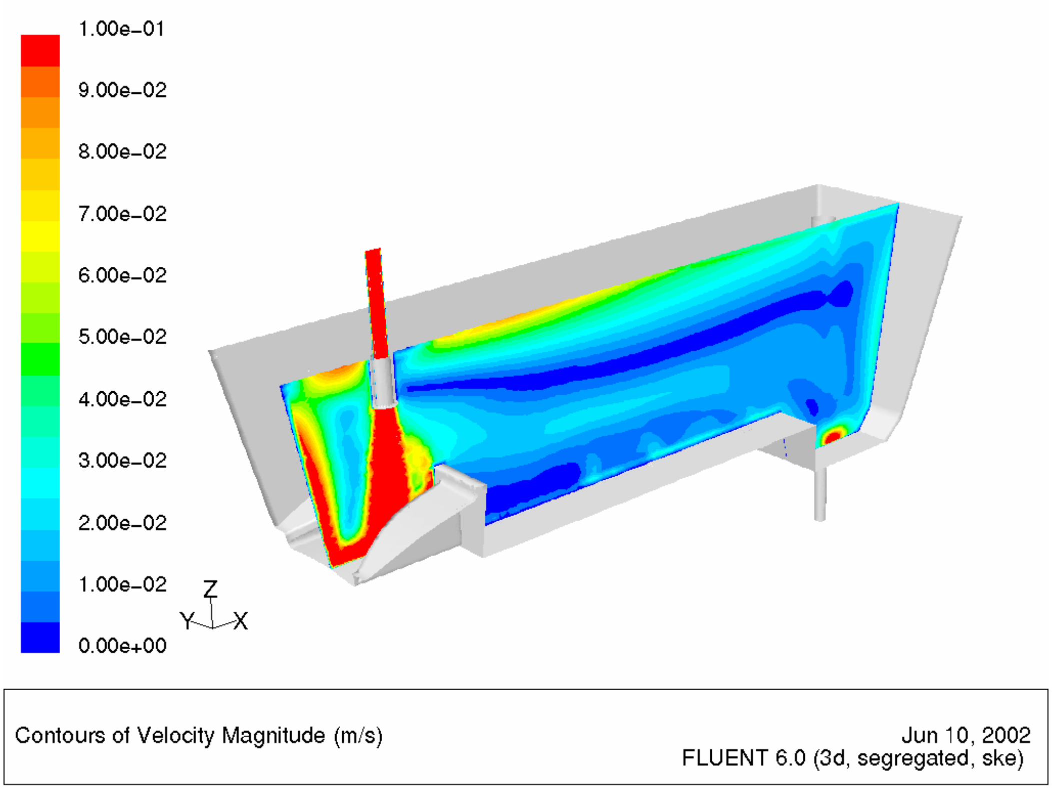

Figure G-11: Velocity magnitude contours on vertical plane – Centre shroud (y=0.32m)

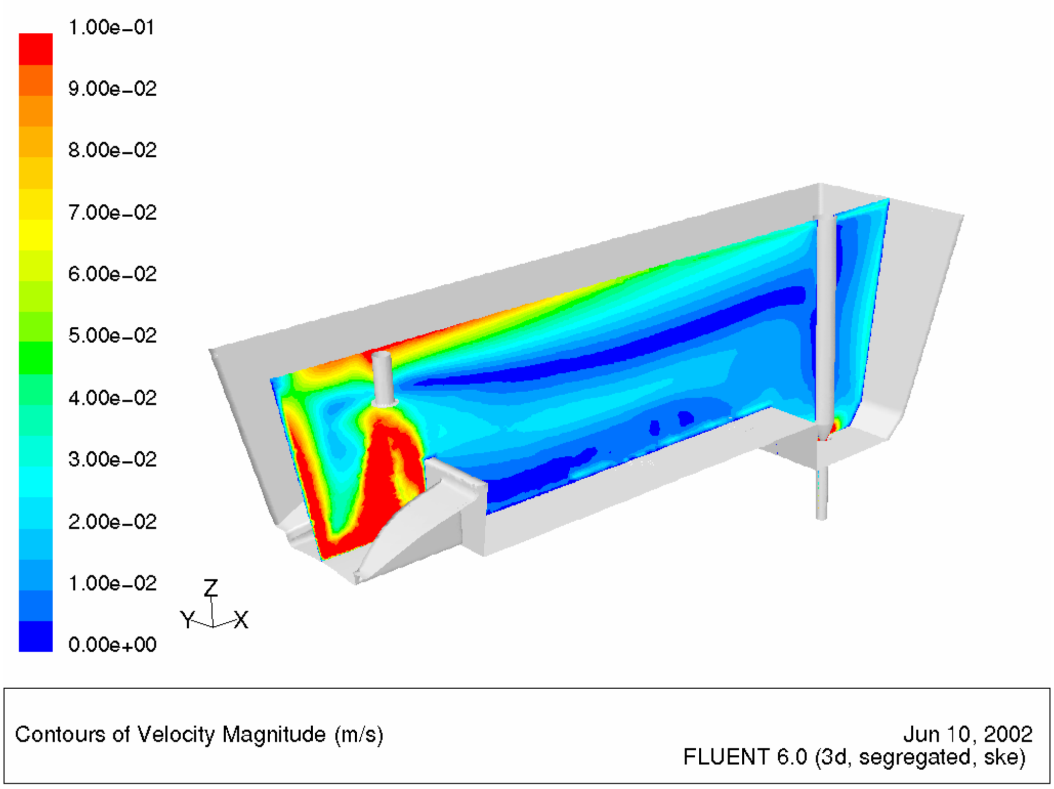

Figure G-12: Velocity magnitude contours on vertical plane – Centre SEN (y=0.4m)

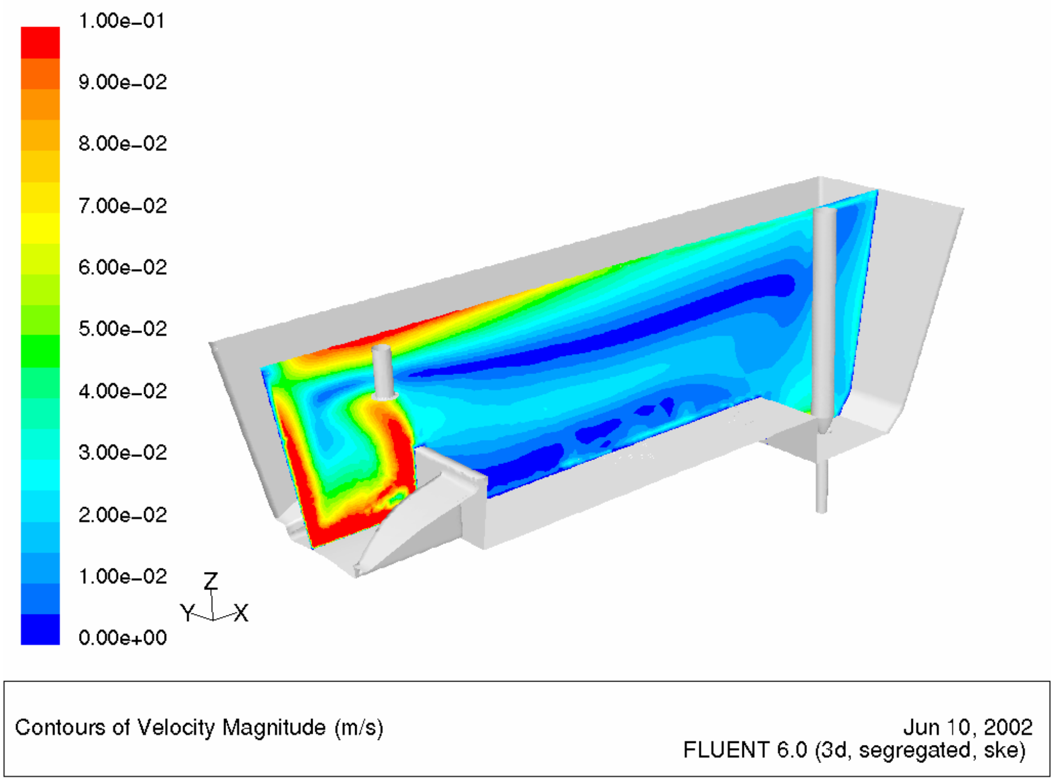

Figure G-13: Velocity magnitude contours on vertical plane – (y=0.48m)

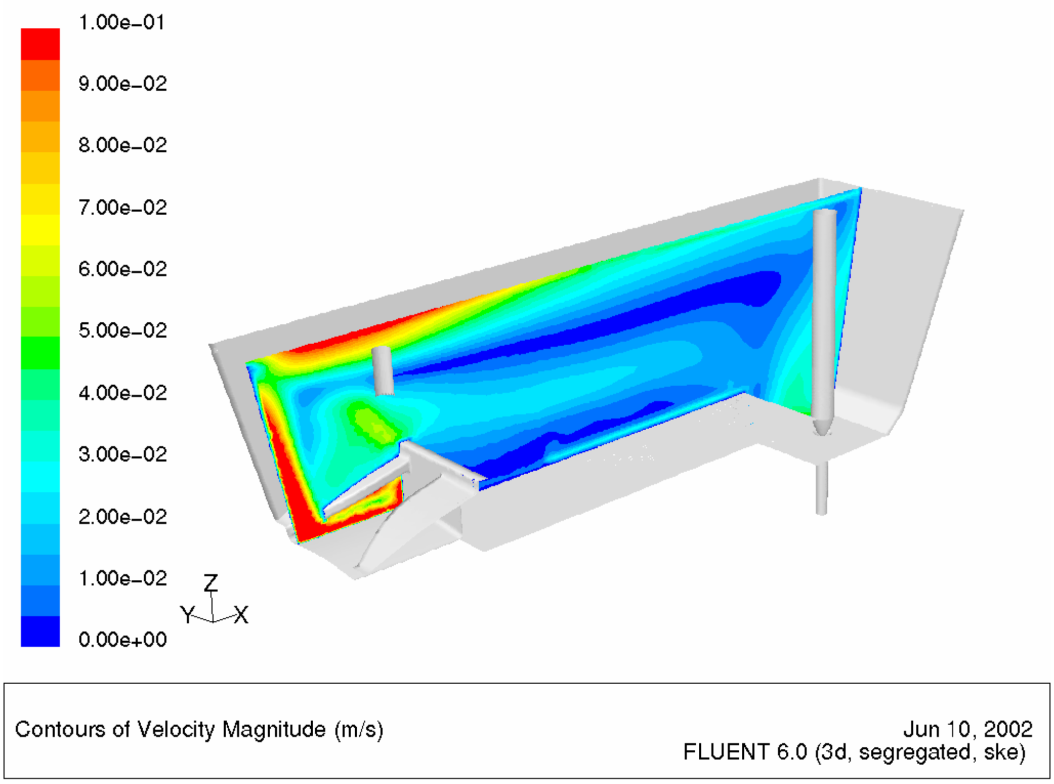

Figure G-14: Velocity magnitude contours on vertical plane – (y=0.6m)

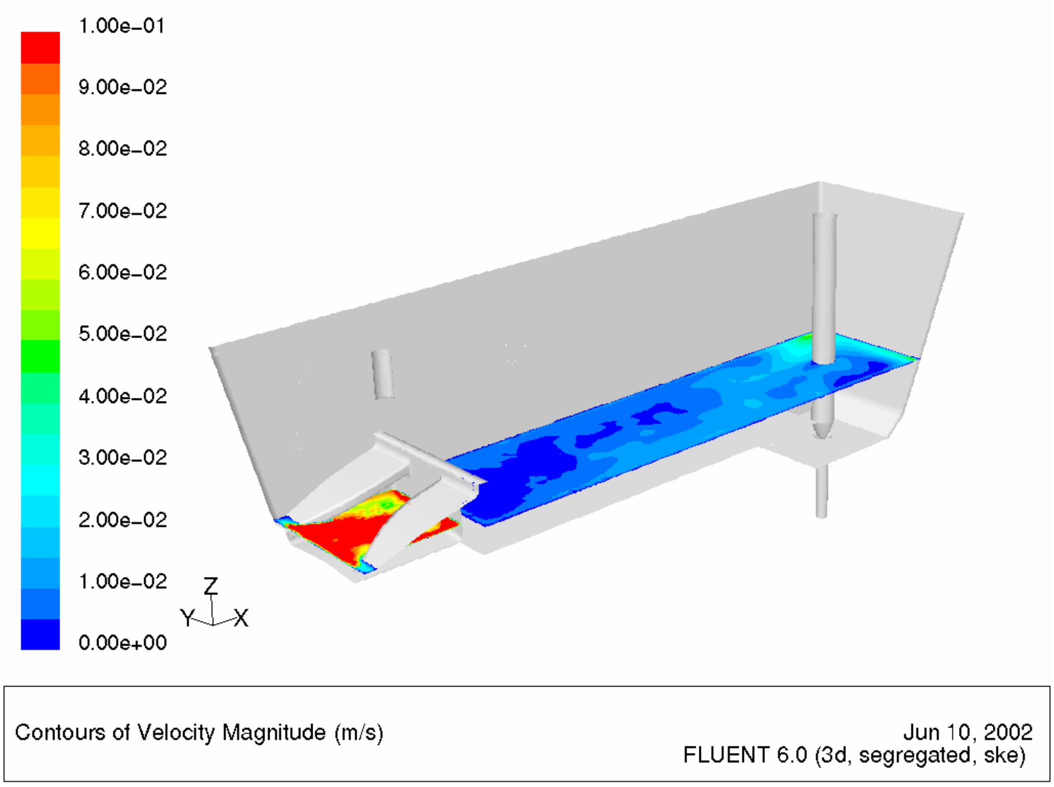

Figure G-15: Velocity magnitude contours on horizontal plane – (z=-0.6m)

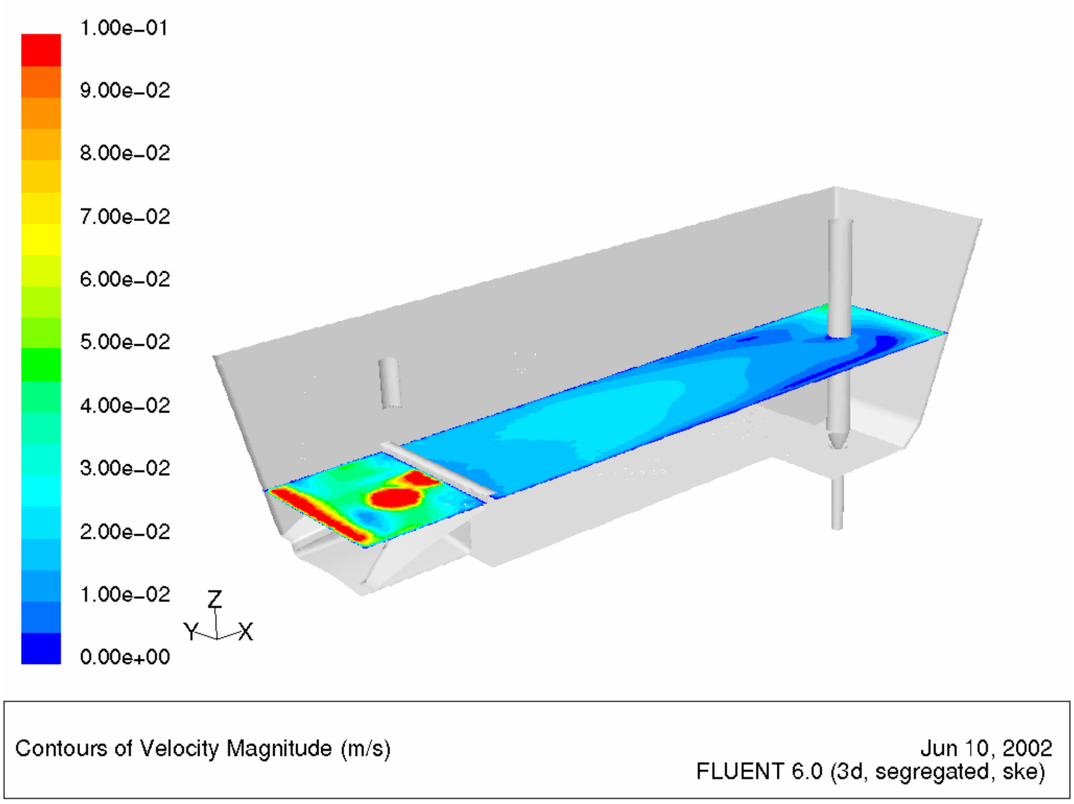

Figure G-16: Velocity magnitude contours on horizontal plane – (z=-0.4m)

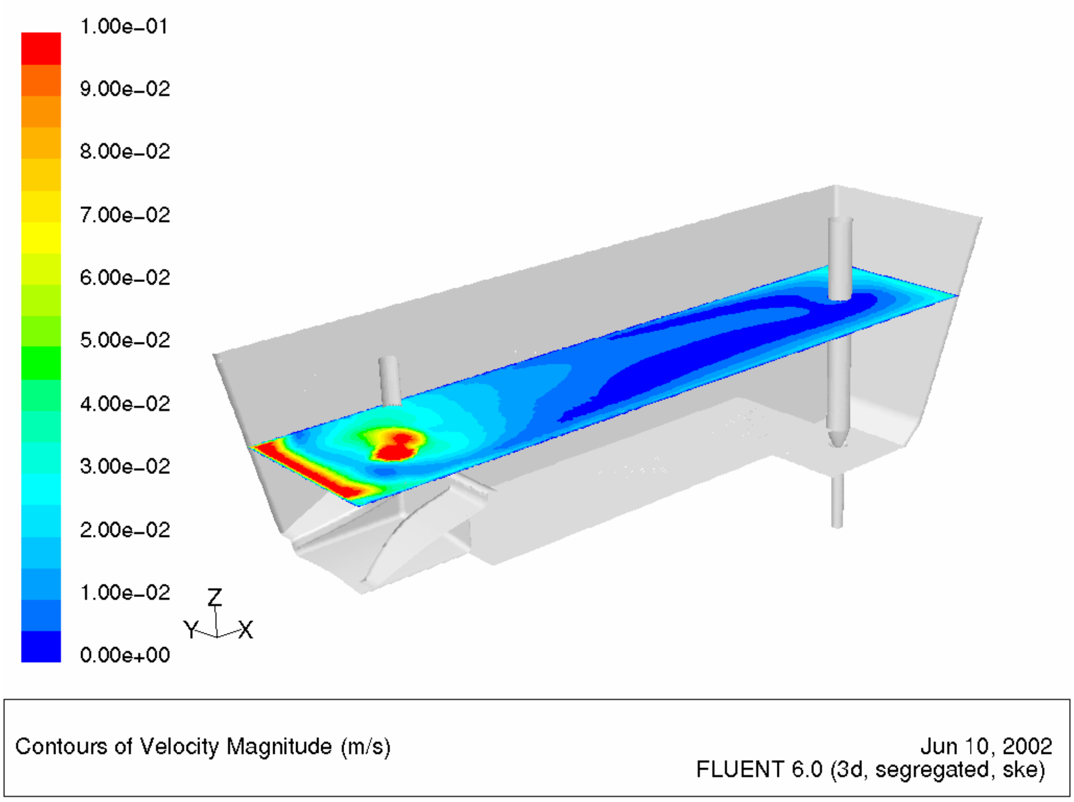

Figure G-17: Velocity magnitude contours on horizontal plane – (z=-0.2m)

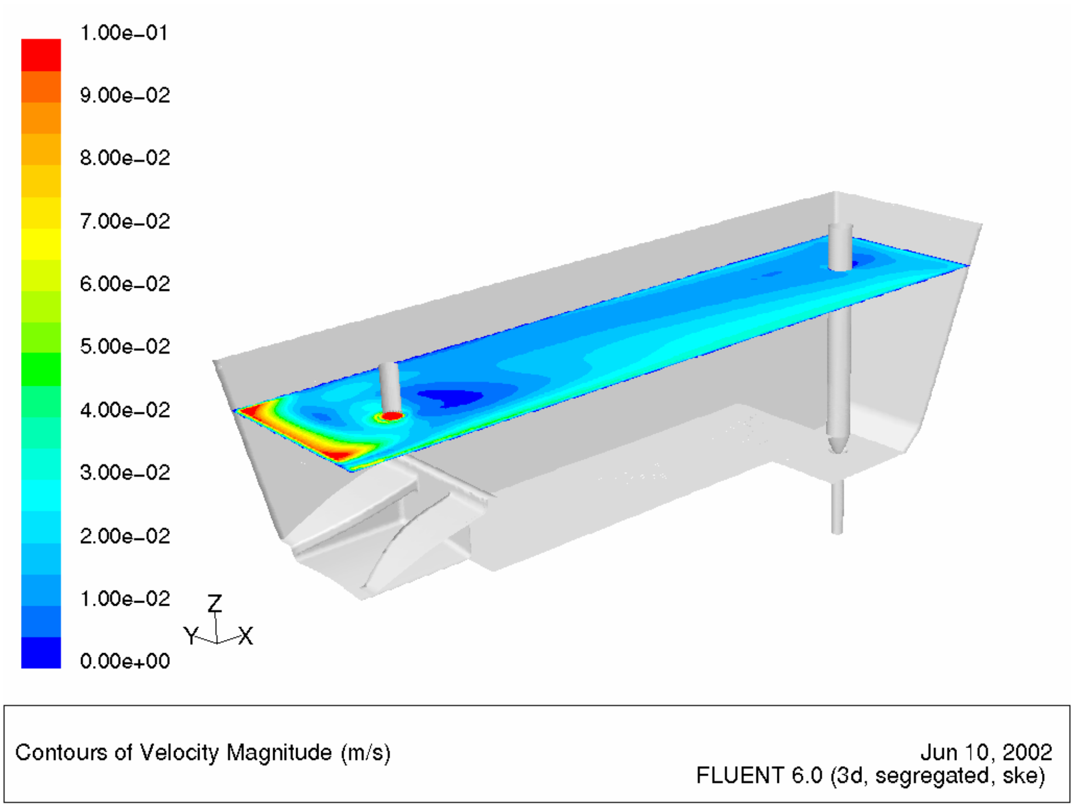

Figure G-18: Velocity magnitude contours on horizontal plane – (z=0.0m)

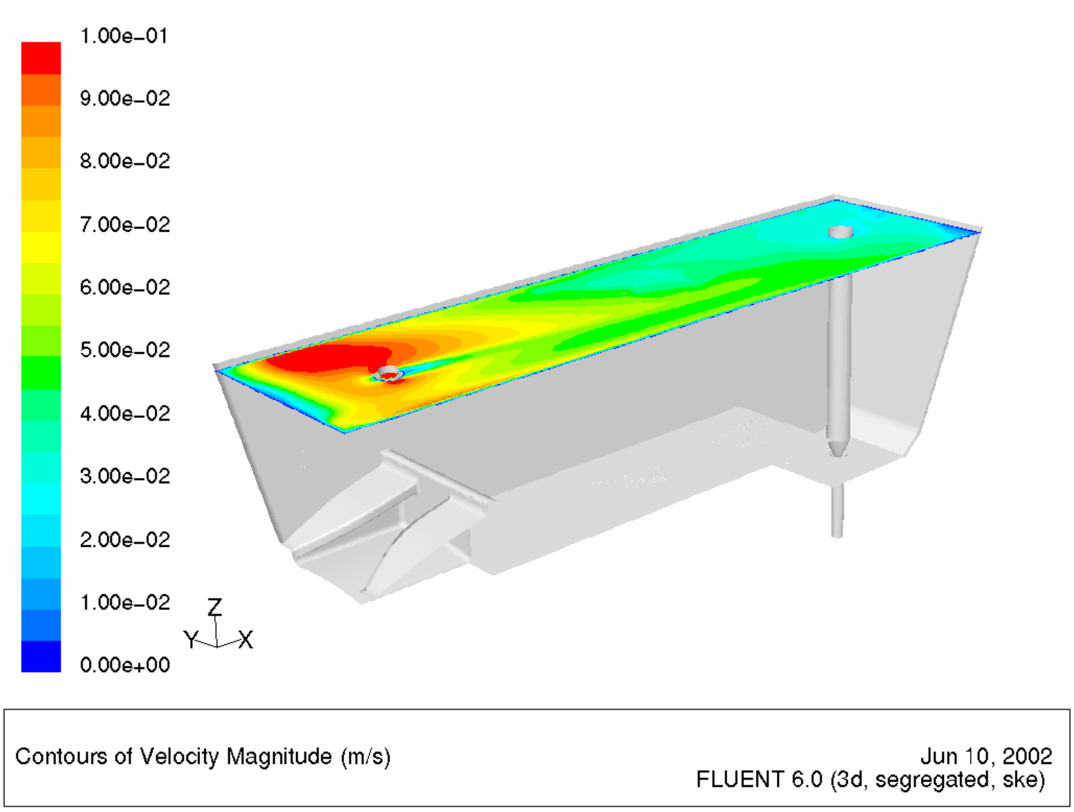

Figure G-19: Velocity magnitude contours on horizontal plane – (z=0.2m)

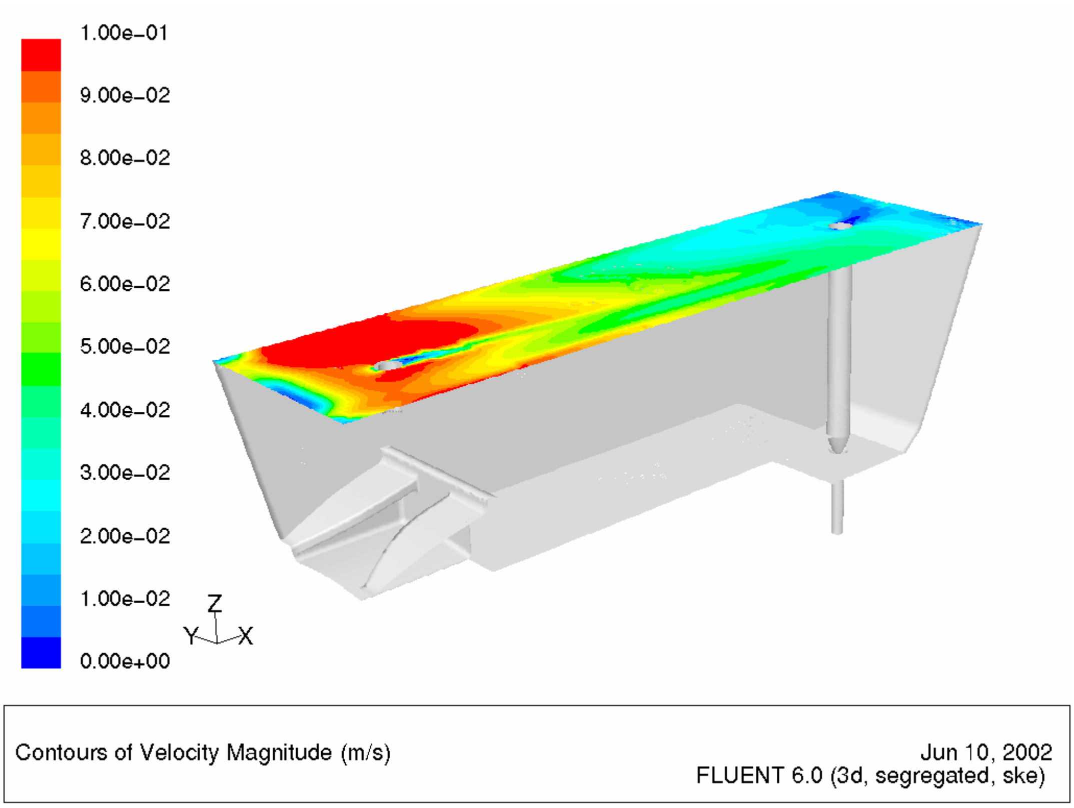

Figure G-20: Velocity magnitude contours on slag layer

G-4 TURBULENT KINETIC ENERGY (TKE)

As the wall TKE can relate to erosion of the refractory, its contours are plotted on all the walls in Figure G-21 and Figure G-22. Note the high levels of TKE on the back wall.

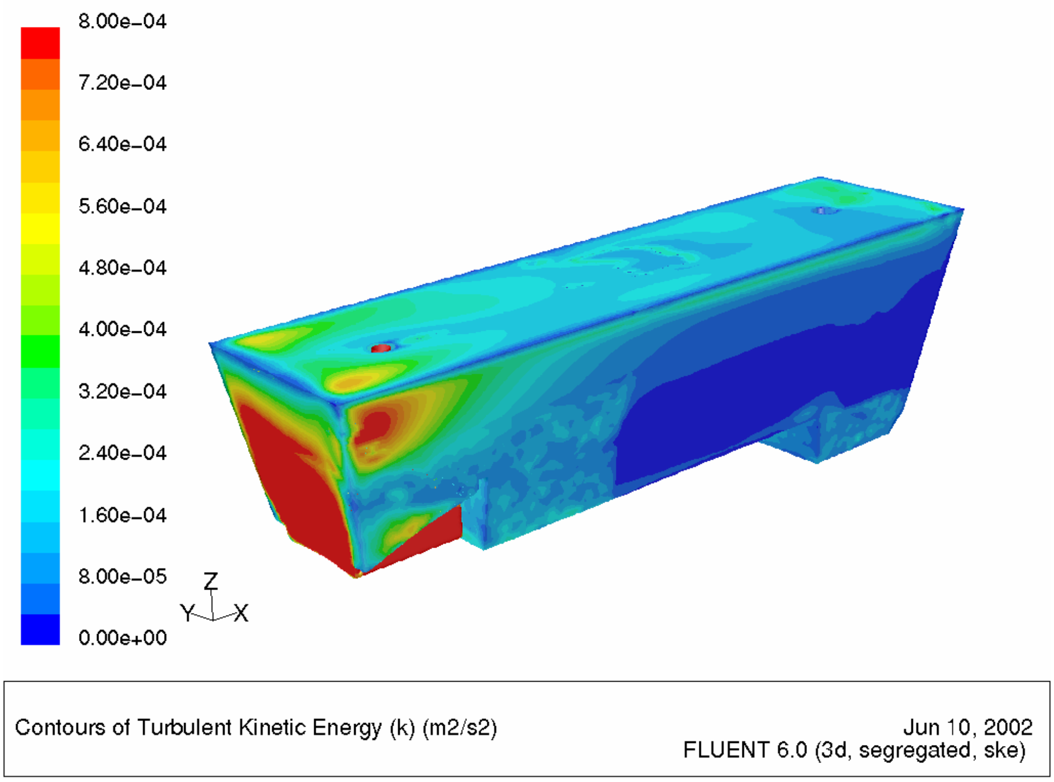

Figure G-21: Wall turbulent kinetic energy contours (front)

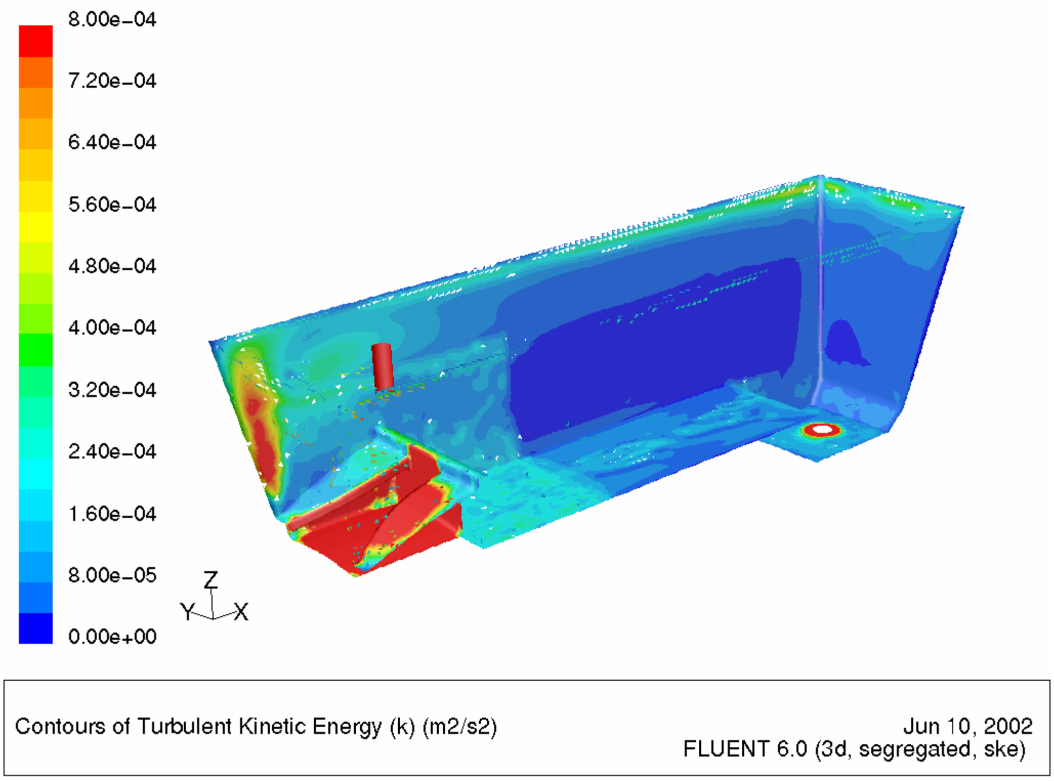

Figure G-22: Wall turbulent kinetic energy contours (back)

Another view of the TKE performance can be seen from selected iso-surfaces as depicted in Figure G-23 through Figure G-25 for three TKE levels.

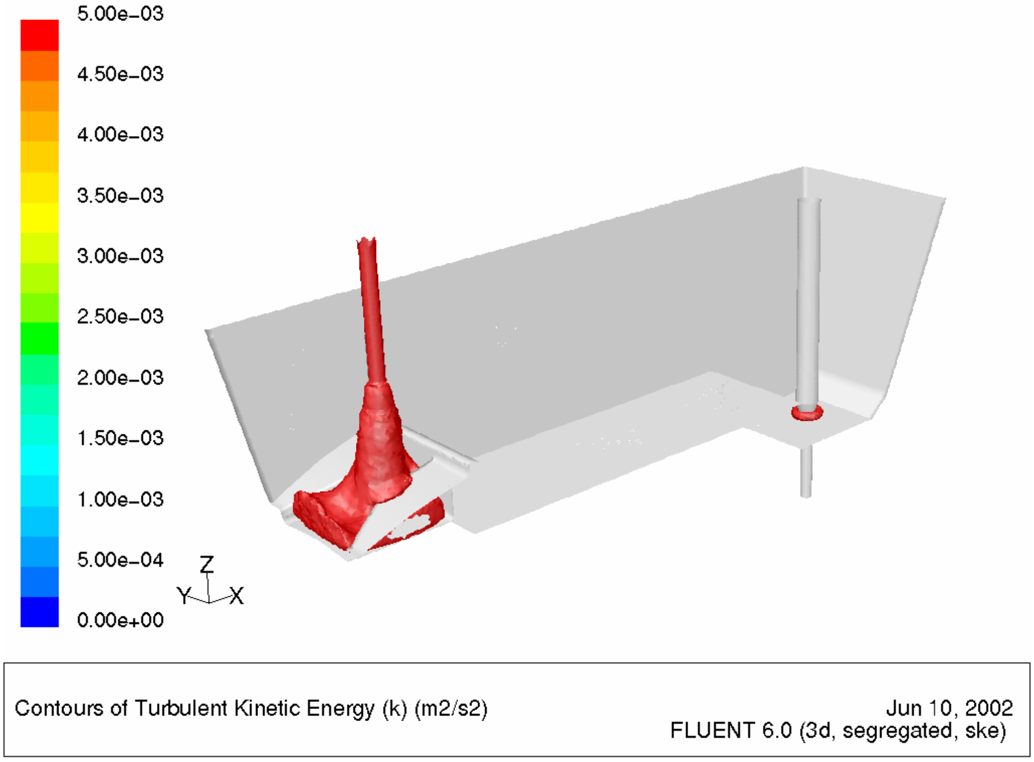

Figure G-23: Iso-surface of turbulent kinetic energy (TKE = 5×10^{-3} m^2.s^{-2})

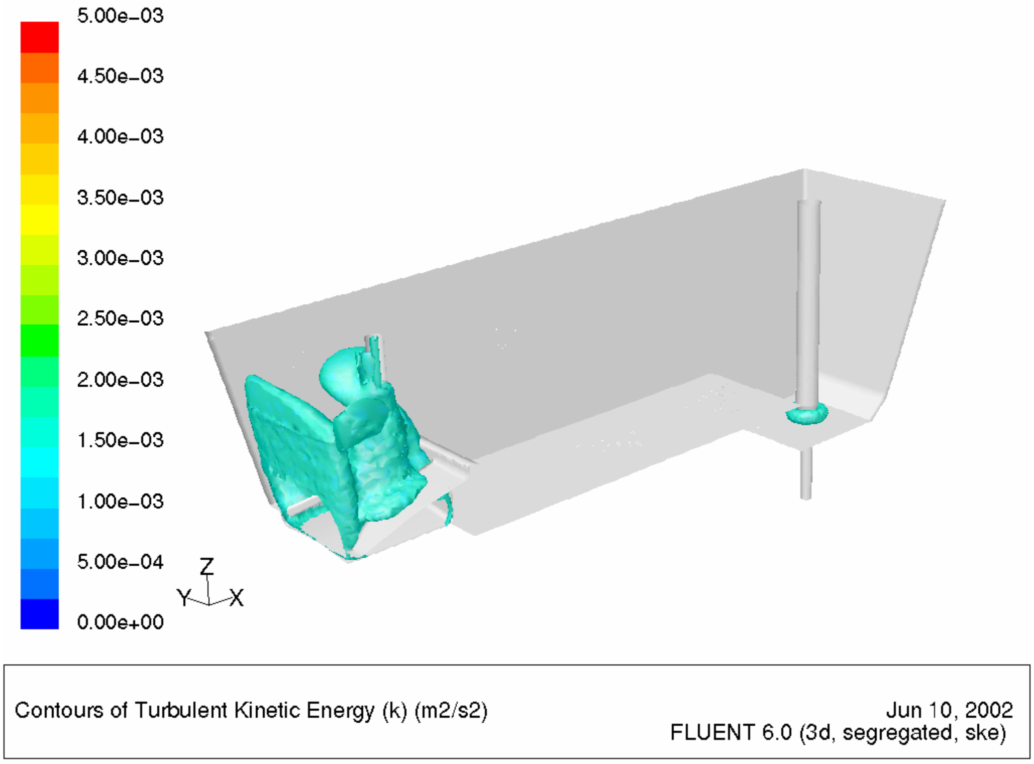

Figure G-24: Iso-surface of turbulent kinetic energy (TKE = 1.5×10^{-3} m^2.s^{-2})

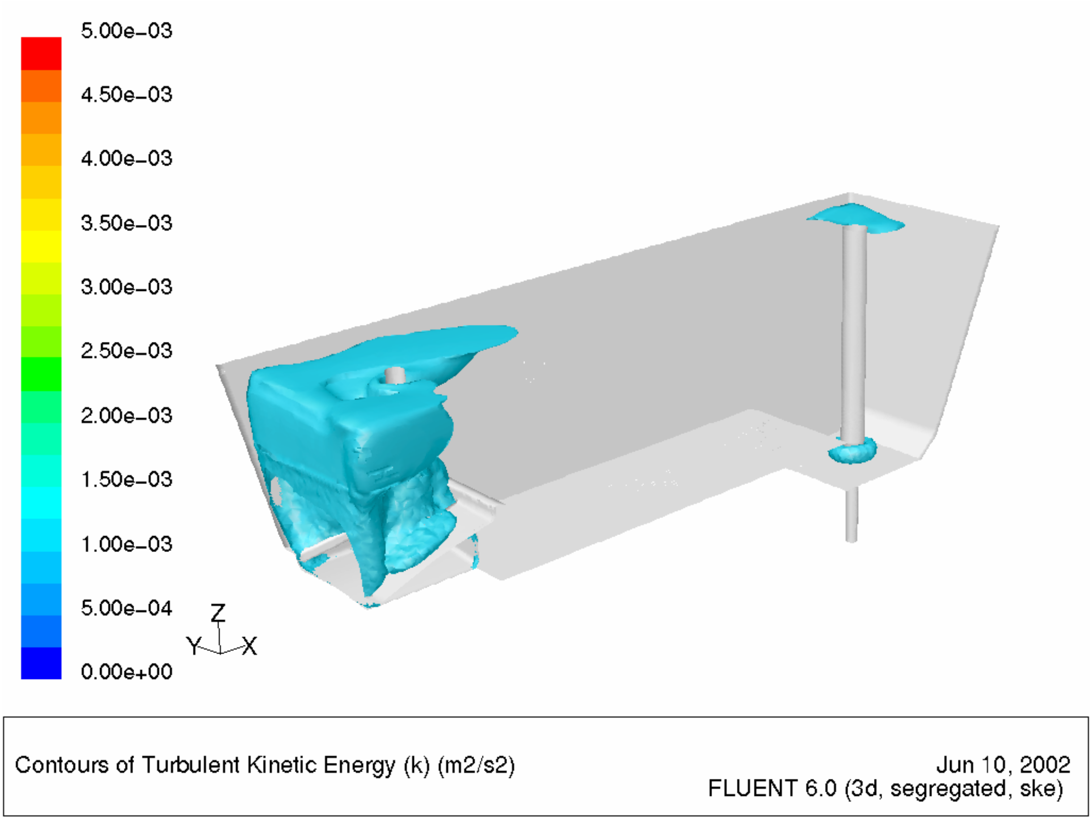

Figure G-25: Iso-surface of turbulent kinetic energy (TKE = 1×10^{-3} $m^2.s^{-2}$)

H

APPENDIX H: PARTICLE TRAPPING USER DEFINED FUNCTION

This appendix contained the first attempt of a more advanced model for the trapping of a particle on the slag layer. This is a C-program that is compiled as a User Defined Function in Fluent. The model calculates the maximum continuous time that a particle spends in a user defined region close to the slag layer. A simple correlation is then used that relates the probability of a particle being trapped to the size of the particle and the time it spent near the slag layer.

```
/************************************************************************/
/*UDF for computing the maximum cont. time spent by a particle in the     */
/* region close to the slag layer                                         */
/* This model assumes:                                                    */
/*    z is pointing upwards                                               */
/* Model inputs:                                                          */
/*    thickness of trapping zone                                          */
/*    z-position of slag layer                                            */
/*                                                                        */
/* File History:                                                          */
/*      DdK 12/2001                                                       */
/*      DdK 4/2002 Automating output of number trapped                    */
/*                                                                        */
/************************************************************************/
```

Appendix H: Particle Trapping User Defined Function

```
#include "udf.h"
#include "sg.h"
#include "dpm.h"

static int flag_inzone;
static int num_trapped;
#define thickness 0.002                /* Thickness that slaglayer's effect into steel */
#define top 0.23                       /* z-coordinate of the top of the surface       */
#define timefact 0.75                  /* Time constant used to determine if trapped   */

DEFINE_INIT(tsns_setup,domain)
{
      /* Allocate particle variables if not been set */
      if (NULLP(user_particle_vars)) Init_User_Particle_Vars();
      /* Set name and lable */
      strcpy(user_particle_vars[0].name,"time-spent");
      strcpy(user_particle_vars[0].label,"Continuous Time Spent at Surface");
      strcpy(user_particle_vars[1].name,"max-time-spent");
      strcpy(user_particle_vars[1].label,"Max. Continuous Time Spent at Surface");
}

DEFINE_DPM_SCALAR_UPDATE(time_spent, cell, thread, initialise, p)
{
      FILE *fp;

      cphase_state_t *c = &(p->cphase);
      if (initialise)
      {
            /* Initialization call */
            flag_inzone = 0;
            p->user[0] = 0;
            p->user[1] = 0;
      }
      else
      {
          /* Calculate time spent in trapping zone (user[0]) and assign to user[1] if longer than current user[1]*/
            if (p->state.pos[2]>(top-thickness) && p->state.pos[2]<(top+thickness))
```

```
                {
                        flag_inzone = 1;
                        p->user[0] += P_DT(p);
                }
                else if (flag_inzone)
                {
                        if (p->user[0]>p->user[1]) p->user[1]=p->user[0];  /* Check if bigger than current max time */
                        flag_inzone = 0;                                   /* Reset flag */
                        p->user[0] = 0;                                    /* Reset counter */
                }
                /*fp = fopen("output.txt","a");
          fprintf(fp, "%f %f %f %f\n", p->state.pos[0], p->state.pos[1], p->user[0], p->user[1]);
          fclose(fp);    */
        }
}

DEFINE_DPM_OUTPUT(time_spent_output, header, fp, p, thread, plane)
{
        /* Post-processing of particle data */
        char name[100];

        if (header)
        {
                num_trapped = 0;
                if (NNULLP(thread))
                        fprintf(fp,"(%s %d)\n",thread->head->dpm_summary.sort_file_name,11);
                else
                        fprintf(fp,"(%s %d)\n",plane->sort_file_name,11);
                fprintf(fp,"(%10s %10s %10s %10s %10s %10s %10s %10s %10s %10s %10s %s)\n",
                                                        "X","Y","Z","U","V","W","diameter","T","mass-flow",
                                                        "time","max-time-spent","name");
        }
        else
        {
                sprintf(name,"%s:%d",p->injection->name,p->part_id);
                fprintf(fp, "((%10.6g %10.6g %10.6g %10.6g %10.6g %10.6g %10.6g"
                                   "%10.6g %10.6g %10.6g %10.6g) %s)\n",
```

```
                                        p->state.pos[0], p->state.pos[1], p->state.pos[2],
                                        p->state.V[0], p->state.V[1], p->state.V[2],
                                        p->state.diam, p->state.temp, p->flow_rate, p->state.time,
                                        p->user[1], name);
                  if (p->user[1] >(timefact*p->state.diam*1e6))
                      {
                       num_trapped = num_trapped + 1;
                       printf("Particle %s leaving %s would have been trapped \n", name,thread->head->dpm_summary.sort_file_name);
                       printf("Total number trapped in trapping zone = %d \n",num_trapped);
                      }
            }
     }
```