Below, the procedure I followed to make some parts of the LSMLIB compatible with MS Visual Studio 2005 is reported. The main goal was to couple the routine that solves the Eikonal equation with the routine that calculates the extension velocity field, for three-dimensional systems. This extension velocity field is an argument for the routine that solves the Eikonal equation.

Step 1: Solve the Eikonal equation (without extension velocity field)

The program file "test_solveEikonalEquation3d.c" to test the "solveEikonalEquation3d"-routine was written, which is very similar to the "test_solveEikonalEquation2d.c" file in LSMLIB in folder "LSMLIB-1.0.0b/examples/serial/fast_marching_method/".

This file was grouped in a project (Eikon_3D) with the following list of files:

FMM_Core.c
FMM_Core.h
FMM_Heap.c
FMM_Heap.h
lsm_FMM_eikonal3d.c
lsm_FMM_eikonal.c (=> "templated" implementation of Eikonal equation solver)
FMM_Callback_API.h
lsm_fast_marching_method.h
FMM_Macros.h
LSMLIB_config.h

In the header file "LSMLIB_config.h" following replacements have been made:

Line 25

```
#define LSMLIB_REAL @LSMLIB_REAL@
```

Replace by

```
#define LSMLIB_REAL double
```

Line 30

```
#define LSMLIB_ZERO_TOL @LSMLIB_ZERO_TOL@
```

Replace by

```
#define LSMLIB_ZERO_TOL 1e-8
```

Line 35

```
#define LSMLIB_REAL_MAX @LSMLIB_REAL_MAX@
```

Replace by

```
#define LSMLIB_REAL_MAX 1.7976931348623158e+308
```

Line 40

```
#define LSMLIB_REAL_MIN @LSMLIB_REAL_MIN@
```

Replace by

```
#define LSMLIB_REAL_MIN 2.2250738585072014e-308
```

When building this project, following error occurred:

```
c:\thomas\lsmlib\lsmlib_v0.9.0\examples\serial\fast_marching_method\eikon_3d\lsm_FMM_eik
onal.c(179) : error C2275: 'FMM_FieldData' : illegal use of this type as an expression
```

This problem disppears when "FMM_FieldData" is replaced by "struct FMM_FieldData" in the file "lsm_FMM_eikonal.c" on the following lines. (It is important to mention that the line references are always the current line numbers before the actual change was made and not the line numbers in the final code.)

Line 94:

```
void FMM_EIKONAL_INITIALIZE_FRONT(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Replace by

```
void FMM_EIKONAL_INITIALIZE_FRONT(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Line 109:

```
LSMLIB_REAL FMM_EIKONAL_UPDATE_GRID_POINT_ORDER1(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Replace by

```
LSMLIB_REAL FMM_EIKONAL_UPDATE_GRID_POINT_ORDER1(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Line 128:

```
LSMLIB_REAL FMM_EIKONAL_UPDATE_GRID_POINT_ORDER2(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Replace by

```
LSMLIB_REAL FMM_EIKONAL_UPDATE_GRID_POINT_ORDER2(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Line 179 and line 180 (twice):

```
 FMM_FieldData *fmm_field_data =
    (FMM_FieldData*) malloc(sizeof(FMM_FieldData));
  if (!fmm_field_data) return LSM_FMM_ERR_FMM_DATA_CREATION_ERROR;
  fmm_field_data->phi   = phi;
  fmm_field_data->speed = speed;
```

Replace by

```
  struct FMM_FieldData *fmm_field_data =
    (struct FMM_FieldData*) malloc(sizeof(struct FMM_FieldData));
  if (!fmm_field_data) return LSM_FMM_ERR_FMM_DATA_CREATION_ERROR;
  fmm_field_data->phi   = phi;
  fmm_field_data->speed = speed;
```

Line 259:

```
void FMM_EIKONAL_INITIALIZE_FRONT(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Replace by

```
void FMM_EIKONAL_INITIALIZE_FRONT(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Line 312:

```
LSMLIB_REAL FMM_EIKONAL_UPDATE_GRID_POINT_ORDER1(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Replace by

```
LSMLIB_REAL FMM_EIKONAL_UPDATE_GRID_POINT_ORDER1(
```

```
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Line 446:

```
LSMLIB_REAL FMM_EIKONAL_UPDATE_GRID_POINT_ORDER2(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Replace by

```
LSMLIB_REAL FMM_EIKONAL_UPDATE_GRID_POINT_ORDER2(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

When building this resulting project, following errors occurred:

```
1>c:\thomas\lsmlib\lsmlib_v0.9.0\examples\serial\fast_marching_method\eikon_3d\lsm_FMM_e
ikonal.c(179) : error C2143: syntax error : missing ';' before 'type'
1>c:\thomas\lsmlib\lsmlib_v0.9.0\examples\serial\fast_marching_method\eikon_3d\lsm_FMM_e
ikonal.c(181) : error C2065: 'fmm_field_data' : undeclared identifier
1>c:\thomas\lsmlib\lsmlib_v0.9.0\examples\serial\fast_marching_method\eikon_3d\lsm_FMM_e
ikonal.c(182) : error C2223: left of '->phi' must point to struct/union
1>c:\thomas\lsmlib\lsmlib_v0.9.0\examples\serial\fast_marching_method\eikon_3d\lsm_FMM_e
ikonal.c(183) : error C2223: left of '->speed' must point to struct/union
1>c:\thomas\lsmlib\lsmlib_v0.9.0\examples\serial\fast_marching_method\eikon_3d\lsm_FMM_e
ikonal.c(189) : warning C4047: 'function' : 'FMM_FieldData *' differs in levels of
indirection from 'int'
1>c:\thomas\lsmlib\lsmlib_v0.9.0\examples\serial\fast_marching_method\eikon_3d\lsm_FMM_e
ikonal.c(189) : warning C4024: 'FMM_Core_createFMM_CoreData' : different types for
formal and actual parameter 1
1>c:\thomas\lsmlib\lsmlib_v0.9.0\examples\serial\fast_marching_method\eikon_3d\lsm_FMM_e
ikonal.c(252) : warning C4022: 'free' : pointer mismatch for actual parameter 1
```

These errors disappear when (still in file “lsm_FMM_eikonal.c”), the following block

```
  /*******************************************
   * set up FMM Field Data
   *******************************************/
  struct FMM_FieldData *fmm_field_data =
    (struct FMM_FieldData*) malloc(sizeof(struct FMM_FieldData));
  if (!fmm_field_data) return LSM_FMM_ERR_FMM_DATA_CREATION_ERROR;
  fmm_field_data->phi   = phi;
  fmm_field_data->speed = speed;
```

On line 177 was placed on line 157.
Compiling led to following error:

```
1>.\lsm_FMM_eikonal.c(62) : fatal error C1189: #error :  "lsm_FMM_eikonal: required
macro FMM_NDIM not defined!"
```

To avoid this problem, lines 62-65-68-71-74 in file “lsm_FMM_eikonal.c” were commented.

Compiling the resulting code led to following error:

```
1>.\lsm_FMM_eikonal.c(191) : error C2065: 'FMM_NDIM' : undeclared identifier
```

, although this identifier was already defined in file “lsm_FMM_eikonal3d.c”. This problem was solved by commenting line 20 in file “file lsm_FMM_eikonal3d.c”:

```
//#define FMM_NDIM                                    3
```

and inserting

```
#define FMM_NDIM                                    3
```

on line 47 in file “lsm_FMM_eikonal.c”.

Building this project did not lead to errors nor warnings ☺

In attachment you can find the project in the .zip file “Eikon_3D”. In file “test_solveEikonalEquation3d.c” two different initial 3D fronts were considered, a box and a sphere, together with different source velocity fields.

The MatLab script “plot_Eikon3D_output.m”, also in the .zip file “Eikon_3D”, allows plotting the arrival time function. In Figure 1, slices at z=8, z=9 and z=10 of the arrival time function phi are visualized for a 20*20*20 system.

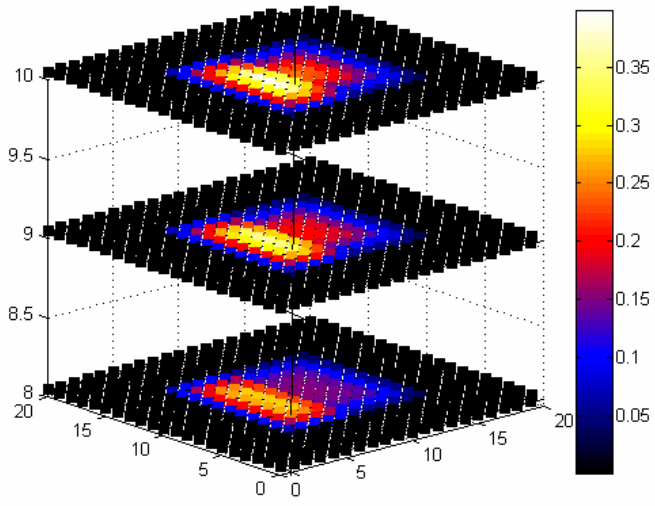

Figure 1: Example of arrival time function obtained with “test_solveEikonalEquation3d.c”.

Step 2: Include extension velocity method

The file "test_Eikonal_ExtensionFields3d.c" was written to test the extension velocity code coupled with the routine to solve the Eikonal equation.

The following files were included to the existing project (= final project of step 1).

"lsm_FMM_field_extension3d.c"
"lsm_FMM_field_extension.c"

The error messages in file "lsm_FMM_field_extension.c" on lines 76, 79, 82, 85, 88, 91 and 94 are commented (otherwise following error message occurred: "`fatal error C1189: #error`").

When building the project, following error appeared:

```
\lsm_FMM_field_extension.c(257) : error C2275: 'FMM_FieldData' : illegal use of this
type as an expression
```

To overcome this problem, "`FMM_FieldData`" was replaced by "`struct FMM_FieldData`" in file "lsm_FMM_field_extension.c":

Line 134:

```
void FMM_INITIALIZE_FRONT_ORDER1(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Replace by

```
void FMM_INITIALIZE_FRONT_ORDER1(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Line 153

```
void FMM_INITIALIZE_FRONT_ORDER2(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Replace by

```
void FMM_INITIALIZE_FRONT_ORDER2(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int num_dims,
```

```
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Line 173:

```
LSMLIB_REAL FMM_UPDATE_GRID_POINT_ORDER1(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Replace by

```
LSMLIB_REAL FMM_UPDATE_GRID_POINT_ORDER1(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Line 194:

```
LSMLIB_REAL FMM_UPDATE_GRID_POINT_ORDER2(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Replace by

```
LSMLIB_REAL FMM_UPDATE_GRID_POINT_ORDER2(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx);
```

Line 257:

```
  FMM_FieldData *fmm_field_data =
    (FMM_FieldData*) malloc(sizeof(FMM_FieldData));
  if (!fmm_field_data) return LSM_FMM_ERR_FMM_DATA_CREATION_ERROR;
  fmm_field_data->phi = phi;
  fmm_field_data->distance_function = distance_function;
  fmm_field_data->num_extension_fields = num_extension_fields;
  fmm_field_data->source_fields = source_fields;
  fmm_field_data->extension_fields = extension_fields;
```

Replace by

```
  struct FMM_FieldData *fmm_field_data =
    (struct FMM_FieldData*) malloc(sizeof(struct FMM_FieldData));
  if (!fmm_field_data) return LSM_FMM_ERR_FMM_DATA_CREATION_ERROR;
  fmm_field_data->phi = phi;
  fmm_field_data->distance_function = distance_function;
  fmm_field_data->num_extension_fields = num_extension_fields;
  fmm_field_data->source_fields = source_fields;
  fmm_field_data->extension_fields = extension_fields;
```

Line 395:

```
void FMM_INITIALIZE_FRONT_ORDER1(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Replace by

```
void FMM_INITIALIZE_FRONT_ORDER1(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Line 653:

```
void FMM_INITIALIZE_FRONT_ORDER2(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Replace by

```
void FMM_INITIALIZE_FRONT_ORDER2(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Line 949:

```
LSMLIB_REAL FMM_UPDATE_GRID_POINT_ORDER1(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Replace by

```
LSMLIB_REAL FMM_UPDATE_GRID_POINT_ORDER1(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Line 1053:

```
LSMLIB_REAL FMM_UPDATE_GRID_POINT_ORDER2(
  FMM_CoreData *fmm_core_data,
  FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Replace by

```
LSMLIB_REAL FMM_UPDATE_GRID_POINT_ORDER2(
  FMM_CoreData *fmm_core_data,
  struct FMM_FieldData *fmm_field_data,
  int *grid_idx,
  int num_dims,
  int *grid_dims,
  LSMLIB_REAL *dx)
```

Building this project led to the following error message:

```
.\lsm_FMM_field_extension.c(259) : error C2065: 'fmm_field_data' : undeclared identifier
```

This error disappeared when moving the following block of code from line 255 in file "lsm_FMM_field_extension.c" to line 94 in file "FMM_Core.h" (Again, these are always the current line positions, before the change was made).

```
/*================ lsm_FMM_field_extension Data Structures
=============*/
struct FMM_FieldData {
  LSMLIB_REAL *phi;                  /* original level set function
(input) */
  LSMLIB_REAL *distance_function;    /* distance function (output)
*/

  int num_extension_fields;          /* number of extension fields
*/
  LSMLIB_REAL **source_fields;       /* source fields to extend off of
zero */
                                     /* level set (input)
*/
  LSMLIB_REAL **extension_fields;    /* computed extension field
(output)   */
```

```
  /* data arrays used for initializing and updating extension fields
*/
  LSMLIB_REAL *extension_fields_numerator;
  LSMLIB_REAL *extension_fields_denominator;
  LSMLIB_REAL *extension_fields_cur;
  LSMLIB_REAL *extension_fields_sum_div_dist_sq;
  LSMLIB_REAL *extension_fields_minus;
  LSMLIB_REAL *extension_fields_plus;
  LSMLIB_REAL *speed;       /* speed function               */
};
```

This error is probably a result of the fact that all declarations need to be done on top of a function or file.

Building this version of the project led to following error:

```
\lsm_FMM_field_extension.c(294) : error C2065: 'FMM_NDIM' : undeclared identifier
```

This problem can be solved in the same way as was done above in **step 1**, i.e.:

Comment line 21 in file "lsm_FMM_field_extension3d.c"

```
#define FMM_NDIM                        3
```

and add this line on line 75 in file "lsm_FMM_field_extension.c".

In the file "test_Eikonal_ExtensionFields3d.c", following block of code was included on Line 149:

```
  for (idx = 0; idx < num_gridpts; idx++) {
      speed[idx]=ext_fields[0][idx];
  }

  /* Carry out FMM calculation */

solveEikonalEquation2d(phi,speed,mask,spatial_derivative_order,grid_di
ms,dx);
```

So the calculated extension velocities field is passed as an argument (i.e. ***speed***) to the function `solveEikonalEquation2d`.

In order to do so, the variable speed was introduced, and had to be declared and identified. This was done in following sections of the project:

In file "test_Eikonal_ExtensionFields3d.c" on…

Line 33:

```
int main( int argc, char *argv[])
{
  /* field variables */
  double *phi;
  double *distance_function;
  double **source_fields;
  double **ext_fields;
```

```
  double *mask = 0;
  double *speed;
```

Line 72:

```
  /* allocate memory for field data */
  num_ext_fields = 1;
  phi = (double*) malloc(num_gridpts*sizeof(double));
  speed = (double*) malloc(num_gridpts*sizeof(double));
```

Line 149:

```
  for (idx = 0; idx < num_gridpts; idx++) {
      speed[idx]=ext_fields[0][idx];
  }
```

Line 165:

```
free(speed);
```

Building this project did not lead to errors nor warnings ☺

In attachment you can find the project in the .zip file "Ext+Eikon_3D". You can also find the MatLab file "plot_Ext+Eikon_3D_output.m" which allows plotting the source velocity field, extensions velocity field and arrival time function.

The method was tested for a 40*40*40 grid, where the initial front was determined as a sphere with centre (20,20,20) and a radius of 15.6. Figure 2 shows the arbitrarily chosen source velocity field. In the following figures, three slices in the 3D domain are visualized, i.e. those at z=8, z=9 and z=10.

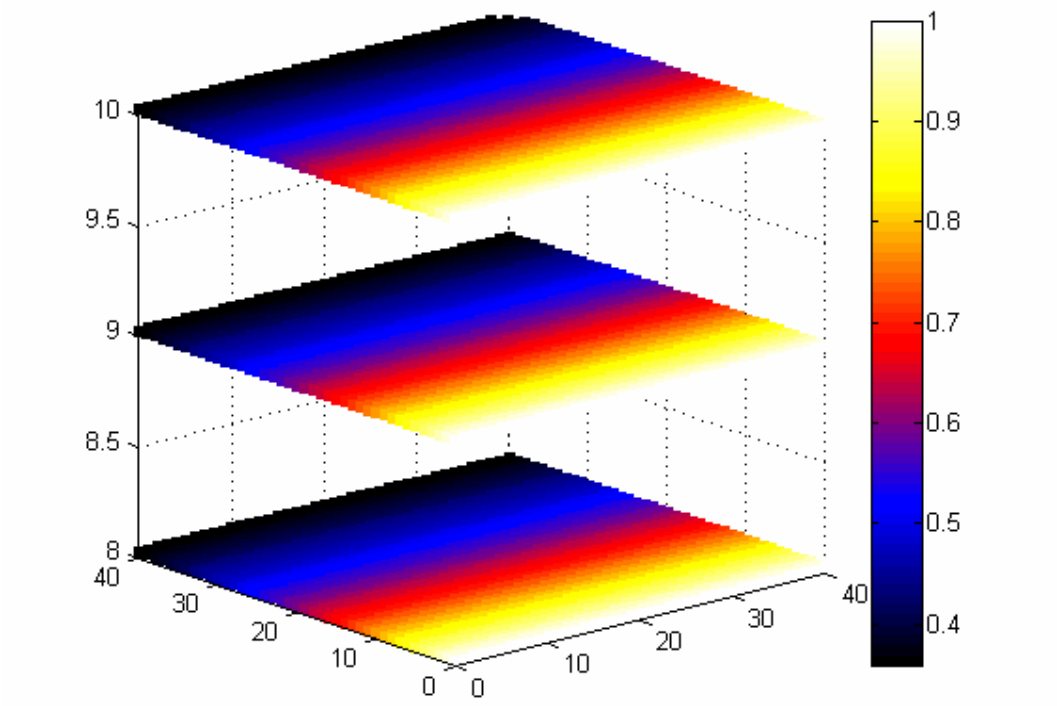

Figure 2: Arbitrarily chosen source velocity field.

Following results for the extension velocity field (Figure 3) and the propagating front of arrival time functions phi (Figure 4) were obtained.

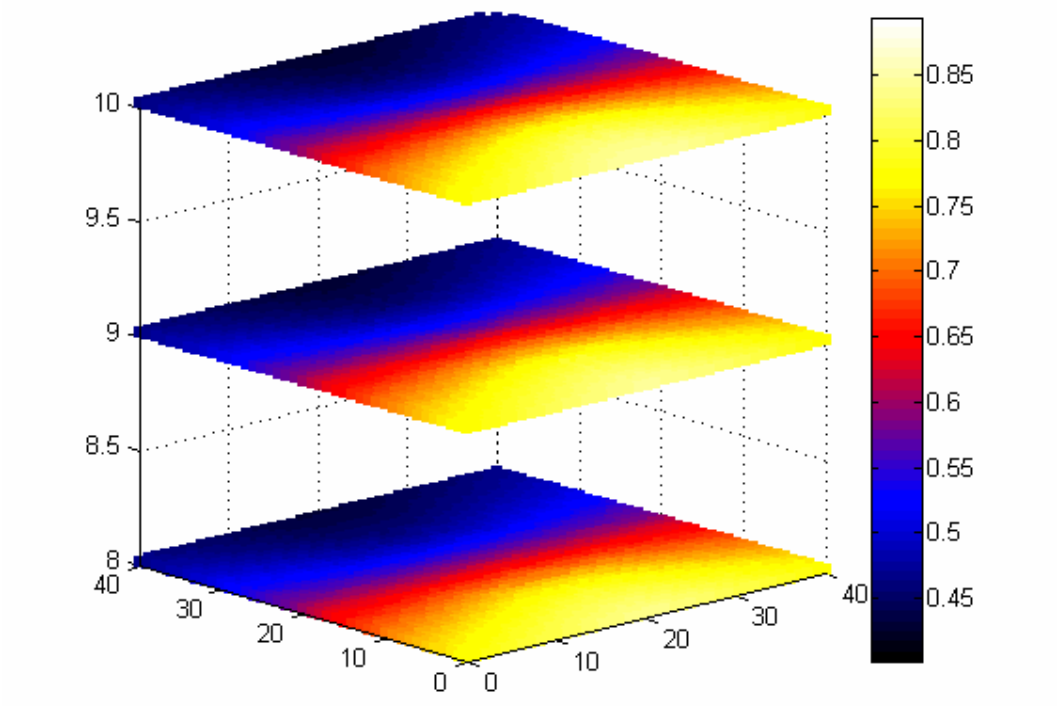

Figure 3: Extension velocity field.

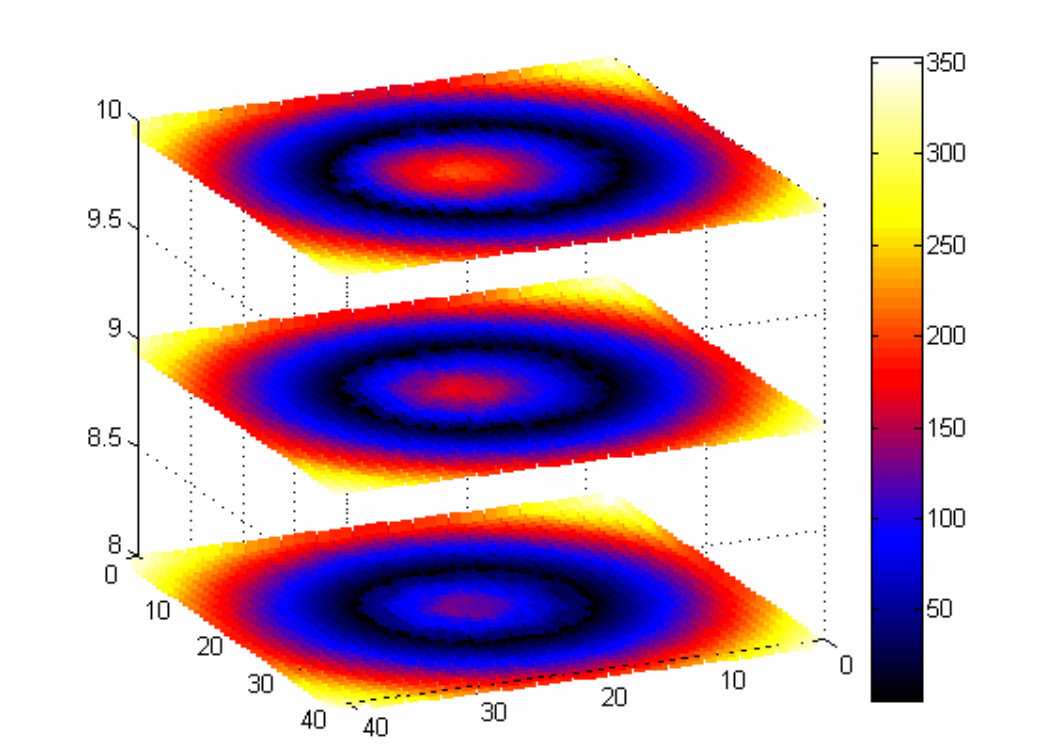

Figure 4: Arrival time function phi. The initial front was a sphere with centre (20,20,20) and with radius of 15.6, in a 40*40*40 grid.