1 Installation

dune-mongeampere has the following dependencies

- cimg header
- eigen3 library
- Umfpack
- boost program options
- adolc
- opennurbs

If those are not found in standard paths, further search paths may be added in the files in dune-mongeampere/cmake/modules.
I added my current version of adolc, opennurbs and cimg. For their installation see ADOLC-C-2.5.2./README and opennurbs/README, respectively .
After all dependecies are installed, you execute the following command in the folder dune-common to execute cmake and make in all dune modules (compare dune installation)

```
@../dune-common/bin: ./dunecontrol configure
@../dune-common/bin: ./dunecontrol all
```

Optionally you may configure install folder and compilter in an option file - compare the dune website and the help of dunecontrol on the option --opts=FILE. If the compilation of some dune modules fails, it is possible to reinit the compilation process of a single target with the parameter --only=dune-MODULENAME. Note that all included dune modules need to be installed. If the configure command was executed successfully, the compile command "make"may also be called in the local build directores.

1.1 Quickstart

After a successful build you should find the program dune_mongeampere_image_OT in the src folder of the building directory. It is called with two further configuration files. The control parameters of the finite element (FE) solver are given by the .ini file which is passed iwth the command line option -c. The configuration ini file of the problems geometry however is passed by the command line option -g. In the folder inputData are two of example init files given
Before calling the program it has to be secured that all output folders exist. E.g. in the case of the example ini files .

```
@..dune-mongeampere/build-cmake: mkdir data
@..dune-mongeampere/build-cmake: mkdir data/PS12Splines
@..dune-mongeampere/build-cmake: mkdir data/PS12Splines/OT
@..dune-mongeampere/build-cmake: mkdir plots
@..dune-mongeampere/build-cmake: mkdir plots/PS12Splines
@..dune-mongeampere/build-cmake: mkdir plots/PS12Splines/OT
```

In the standard build folder you can call the program then with

```
@..dune-mongeampere/build-cmake/src: ./dune_mongeampere_image_OT
-c ../../inputData/SolversquareToSquareOT.ini -g ../../inputData/imageOTsetting.ini
```

If the build folder was chosen manually, the paths need to be adapted accordingly.

2 Programs

dune_mongeampere	solves standard MA equations
dune_mongeampere_OT	solves the in Optimal Transport (OT) arising MA equation $\det(D^2(u) - \frac{f}{g\nabla u} = 0)$ with the sensible transport boundary conditions. Input distributions f and g are fixed in the source code.
dune_mongeampere_image_OT	solves the in Optimal Transport (OT) arising MA equation $\det(D^2(u) - \frac{f}{g\nabla u} = 0)$ with the sensible transport boundary conditions. Input of the distributions f and g are image files
dune_mongeampere_reflector	calculates a reflective surface considering a point light source
dune_mongeampere_reflector_parallel	calculates a reflective surface considering parallel light
dune_mongeampere_refractor	calculates a refractive surface considering a point light source
dune_mongeampere_refractor_parallel	calculates a refractive surface considering parallel light
read_and_write_OT	possibility to read the FE solution given its coefficients, in order to generate plot output
export_refractor_points	generates points on a cartesian grid given a FE solution
fit_surface	fitted a NURBS surface given a cartesian point grid
convert_dat_to_bmp	convertes a .data file to a .bmp file

3 Using dune_mongeampere_...

All necessary input files are listed by the command

```
@../dune-mongeampere/build-cmake/src: ./dune_mongeampere_... -h
```

Usually one configuration file for the geometry, passed by -g and one file for the FE solver, passed by -c, is required.

3.1 Configuration

Note that the some configurations have to be chosen during the compile time. One may differ

1. choice of the finite element space
2. using automatic differentiation
3. parallel light or point source
4. optimisation in case that the source domain Ω is rectangular

For more information see appendix A. All other parameters are given during the run time by the ini files.

3.1.1 Parameter in the FE solver ini file (passed with -c)

solver	initValueFromFile	indicates if the initialguess is read from file
solver	initValue	path to initialguess (given in ASCII coefficient values)
solver	startlevel	start refinement of initial grid
solver	nonlinearSteps	steps of refinement
solver	maxSteps	maximum number of steps in a nonlinear step
solver	Dirichlet	indicate Dirichlet boundary conditions (deprecated)
solver	JacSimultaneously	indicate combined evaluation of function and Jacobian
solver	lambda	penalisation value for determinant
solver	sigma	penalisation value for jumps in DG method (deprecated)
solver	sigmaGrad	penalisation value for gradient jumps in ansatz (deprecated)
solver	sigmaBoundary	penalisation value for weakly forced boundary conditions (deprecated)
dogleg	iradius	initial trust region radius
dogleg	stopcriteria0	$\|f'\|_{inf} \leq stopcriteria(1)$
dogleg	stopcriteria1	$\|dx\|_2 <= stopcriteria(2) * (stopcriteria(2) + \|x\|_2)$
dogleg	stopcriteria2	$\|f\|_{inf} <= stopcriteria(3)$
dogleg	silentmode	suppress output of dogleg solver
dogleg	exportJacobianIfSingular	activate matlab export to std::cout in case of singular Jacobian
dogleg	check_Jacobian	activate a check of the Jacobian with Finite Differences (computationally expensive)
dogleg	exportFDJacobianifFalse	activate matlab export to std::cout in case of wrong Jacobian
output	directory	output folder for data as coefficients
output	plotdirectory	output folder for output data as vtks, rayTracer files, 3dm files ...
output	prefix	output prefix
output	refinement	specify additional grid refinement for plots
output	cartesianGridN	specify the elements for a cartesian export grid
output	write_vtk	write solution to vtk-ASCII files

3.1.2 Parameter of the geometry ini file (passed with -g)

geometry optic	xMin	lower left x value of optical surface
geometry optic	xMax	upper right x value of optical surface
geometry optic	yMin	lower left y value of optical surface
geometry optic	yMax	upper right y value of optical surface
geometry optic	gridfile	path to initial grid file (.msh format)
geometry optic	plotgridfile	path to an additional plotgrid file (.msh format)
geometry optic	boundaryN	number of direction in approximation of source boundary
geometry optic	initialOpticDistance	distance between source and opt.surface for the initial guess
geometry target	xMin	lower left x value of targetet
geometry target	xMax	upper right x value of target
geometry target	yMin	lower left y value of target
geometry target	yMax	upper right x value of target
geometry target	gridfile	path to grid of the target (.msh format)
geometry target	target_is_xy_plane	indicates if target is parallel to xy plane (otherwise parallel to xz is assumed)
geometry target	z	z value of the target
geometry target	boundaryN	number of direction in approximation of target boundary
light in	imageName	path to image of source distribution
light out	targetImageName	path to image of target distribution
solver	epsDivide	controls blurring between steps
solver	epsEnd	controls blurring in the final step
solver	minPixelValue	mininmal pixel value the target is brightened up to
povray	cameraAngle	camera angle for ray tracer config file
povray	jitter	jitter for ray tracer config file
povray	nPhotons	number of photons for ray tracer config file
povray	lightSourceRadius	light source radius for ray tracer config file
povray	lightSourceFalloff	fall of of light ray for ray tracer config file
povray	lightSourceTightness	light source tightness for ray tracer config file
povray	lightSourceIntensity	light source intensity for ray tracer config file
povray	writeAperture	indicate whether light source is additionally cropped in ray tracer file

4 output files

Since the FE solver solves the problem on different grid with different target distributions, output is generated of every intermediate step and indicated by a counter. Imagine we are in the second intermediate step
If not specified otherwise in the ini file the solution u is written to a vtk-file called output_folder/EXAMPLE2numericalSolu in xml format. For more information about those plots see B.

4.1 Output in the OT case

Additionally the transport of the source grid Ω, i.e. $nablau(\Omega)$, is written in another .vtk-Datei called output_folder/EXAMPLE2outputGrid.vtu.

4.2 Output of the optical surfaces

The file output_folder/EXAMPLEnumericalSolutionreflector2.pov can be used to render the target under the current optical surface with the ray tracer povray [**?**].
The file output_folder/EXAMPLEnumericalSolutionreflector2.3dm exports a triangular mesh of the current optical surface.
The file output_folder/EXAMPLEnumericalSolutionreflectorPoints2.3dm exports a point cloud of the optical surface.

4.3 Conversion in 3dm NURBS files

Note that output_folder/EXAMPLEnumericalSolutionreflector2.3dm is only an linear approximation on the actual optical surface. To export a better approximation EXAMPLEnumericalSolutionrefractorPoints4.3dm in case of a rectangular grid do the following.

```
@../src: ./export_refractor_points -g ../inputData/Optic/lensSetting.ini
-o ../inputData/Optic/ExportOptions.ini
@../src: ./fit_surface ../plots/Lens/EXAMPLEnumericalSolutionrefractorPoints4.txt
  ../plots/Lens/EXAMPLEnumericalSolutionrefractorPoints4.3dm
```

This generates a Cartesian point cloud of the example and fits a cubic NURBS surface.

A Paramter during compile time

A.1 Choice oft the finite elements

The desired finite element is enabled per a preprocessor variable. Up-to-date is probably only the variable PS12, which enables C^1-continuous S-Splines based on the Powell-Sabin-12-Split.

A.2 Automatic differentiation

Calculations without automatic differentiation by adolc is not possible at the moment ... In the implementation with known derivative (i.e. only in the OT case) there is some bug ...

A.3 Kind of light source

To change the point light source to a source with parallel light, the preprocessor variable PARALLEL_LIGHT has to be defined. This however, is not made manually, but the cmake file compiles both versions separately.

A.4 Rectangular grid

To allow some simplifications and optimisation in case of a rectangular source grid you should define in the file solver/solver_config.h die preprocessor variable RECTANGULAR_GRID.

B Information in the vtk file

output_folder/EXAMPLE2numericalSolution.vtu contains not only the values of u, but also the additional attributes

gradx	∇u
HessianIJ	entries of the Hessian D^2u_{IJ}
Residual	the residual (e.g. $\det(D^2u) - \frac{f}{g(\nabla u)}$ in the OT case)
EV1/2	eigen values of D^2u
Curvature	the Gaussian curvature of the surface (only for in case of optics)

In the OT case output_folder/EXAMPLE2outputGrid.vtu includes the attribut est. integral:
In every triangle Δ of the grid of Ω the integral $\frac{1}{2}\int_\Delta f\,dx$ was estimated by taking the average of f at the corners and multiplying by the area $|\Delta|$. With the OT problem property it holds that the integral $\frac{1}{2}\int_{\tau(\Delta)} g\,dy$ is the same. Hence we the plot stores the value $\frac{1}{2}\int_\Delta f\,dx/\,|\tau(\Delta)|$ which should resemble g for small Δ.