microchannels-optimizer-2D: User Manual

Marcus Hwai Yik Tan

Created on January 26nd, 2016. Last revised on May 15, 2021

1 Objective

This project optimizes channel designs in 2D rectangular domain by combining a 2D polynomial IGFEM solver (in the directory microchannels-IGFEM-2D), sensitivity analysis (Tan et al. [4], Najafi et al. [3]) and gradient-based optimization algorithms in MATLAB. It can do three types of optimization:

1. Standard optimization (Sec. 3.1)
2. Generation of Pareto front (Sec. 3.2)
3. Optimization of 'worst-case scenario' (Sec. 3.3)

2 Compilation

Ensure that microchannels-IGFEM-2D works before trying to run this project. This project requires the compilation of some Mex functions to work. See README in the parent directory for more information.

3 Inputs

The project can be run from either of these main scripts:

1. `main.m`,
2. `main_pareto.m`,
3. `main_blocked_channels.m`,

which respectively call

1. `optimize_channels.m`,
2. `generate_pareto_front.m`,
3. `optimize_blocked_channels.m`

respectively for the three types of optimization listed in Sec. 1.

Many other inputs similar to those in `main.m` of microchannels-IGFEM-2D can also be set in the last three functions. One can also set the optimization algorithm, optimization tolerances, gauss quadrature for the sensitivity analysis (which could be different from that of the FEM analysis) etc. The inputs that can be set by the user are clearly explained and delineated in the functions.

3.1 Standard optimization

Standard optimization minimizes an objective such as p-norm of the temperature, variance of temperature etc subject to non-linear constraints such as pressure drop, maximum temperature and area fraction (Tan et al. [4]). The input arguments for `optimize_channels.m` are described in Table 1 and 2.

Table 1: Input arguments for `optimize_channels.m`.

Argument #	Description
1	Input channel file name (see User_manual in the IGFEM-Curves-2D directory about the channel input file format and the GUI tool used to generate the file)
2	File for imposing geometric constraints to prevent self-intersection (see Sec. 3.4). Set to [ ] if not applicable.
3	Either file with a set of initial designs (see Sec. 3.5) or input channel file if the initial designs are to be generated by choosing random values between the lower and upper bounds. Set to [ ] if the initial design is that given in the input channel file #1
4	Either job number if #4 > 0 or sample number if #4 <0 and #19 == 1. This argument only takes effect when an initial design file (#2) is given. If #4 > 0, the sample number is calculated as (#4 - 1)×#19 + current simulation number, where 1 $\leq$ current simulation number $\leq$ #19.
5	Output directory in the same directory as `main.m`
6	File prefix for the output files
7	Vector of the number of elements in each direction
8	Vector of the domain bounds [xi,xf,yi,yf], where xi,xf,yi,yf are respectively the left, right, bottom and top coordinates of the domain
9	Cost function type. One of the following: `P_NORM` (p-norm of temperature field), `VARIANCE` (variance of temperature field), `PRESSURE` (pressure drop across network, assuming only one inlet and outlet), `AREA` (total area fraction of channels), `TPA` (Weighted combination of p-norm of temperature, pressure drop and area fraction)
10	p of p-norm. Only applicable when #9==`P_NORM`
11	Weights for the cost functions when #9==`TPA`
12	Scales to normalize the cost functions when cost function type is `TPA`

Table 2: Input arguments for `optimize_channels.m`.

Argument #	Description
13	Non-linear constraint types. Any combination of the following strings delimited by a comma: `Pmin` (lower bound on pressure drop) `Pmax` (upper bound on pressure drop), `Tmax` (upper bound on maximum temperature), `Amin` (lower bound on area fraction) `Amax` (upper bound on area fraction), `Vmin` (lower bound on volume fraction) `Vmax` (upper bound on volume fraction). Example: `'Pmax,Tmax'` or `'Tmax,Pmax'` (order immaterial)
14	Lower bound of pressure drop. Only applicable when #13 contains `'Pmin'`
15	Upper bound of pressure drop. Only applicable when #13 contains `'Pmax'`
16	Upper bound of maximum temperature. Only applicable when #13 contains `'Tmax'`
17	Lower bound of area/volume fraction. Only application when #13 contains `'Amin'`/`'Vmin'`
18	Upper bound of area/volume fraction. Only application when #13 contains `'Amax'`/`'Vmax'`
19	Number of simulations per job. For each job, #19 simulations will be run.
20	Logical: true (use the first initial design #3), false (use the network in #1)
21	Logical: true (merge triangles used for geometrical constraints to form quadrilaterals)
22	'3D' conductivity
23	Domain thickness for calculating '2D' conductivity and volume fraction
24	Convection coefficient h_{conv} of $h_{conv}(T - T_{amb})$
25	Ambient temperature T_{amb} of $h_{conv}(T - T_{amb})$
26	Uniform heat source value. If analytical distributed heat source needs to be provided, use the two files `body_source_functions.cpp` in the directories mx_FEM and mx_sensitivity. The spatial derivative of the distributed heat source needs to be provided to the file located in mx_sensitivity. It is the responsibility of the user to ensure that the analytical forms in both files are consistent. In the mx_FEM directory, one must uncomment the directives `#define source_functions` in `compute_IGFEM_element.cpp` and `compute_regular_element.cpp`. Also, the same directive is to be uncommented in `IGFEM_element_pseudo_adjoint_forces.cpp`

3.2 Generation of Pareto front

Pareto fronts involving up to three objectives are generated using the normalized normal constraint method (Messac et al. [2]). Objectives available are p-norm of temperature, pressure drop and area fraction. The input arguments for `generate_pareto_front.m` are described in Table 9 and 4.

Table 3: Input arguments for `generate_pareto_front.m`.

Argument #	Description
1	Input channel file name (see User_manual in the IGFEM-Curves-2D directory about the channel input file format and the GUI tool used to generate the file)
2	File for imposing geometric constraints to prevent self-intersection (see Sec. 3.4). Set to [ ] if not applicable.
3	Output directory in the same directory as `main_pareto.m`
4	File prefix for the output files
5	Vector of the number of elements in each direction
6	Vector of the domain bounds [xi,xf,yi,yf], where xi,xf,yi,yf are respectively the left, right, bottom and top coordinates of the domain
7	Cost function type. Any combination of the following strings delimited by a comma: `T` (p-norm of temperature field), `P` (pressure drop) and `A` (area fraction). Example: `'T,P'` means that the Pareto front with the p-norm and pressure drop as the first and second objectives, respectively are to be generated. If it is `'P,T'`, the first objective is the pressure drop.
8	p of p-norm. Only applicable when #7 contains `P`
9	Matrix of unnormalized anchor points with the i-th column being the objective values corresponding to the minimization of the i-th objective *alone*. Example: say #7== `'T,P'`, the objective values corresponding to the minimization of the p-norm alone are respectively $T^{(T)}$ and $P^{(T)}$ and those corresponding to the minimzation of the pressure drop alone are respectively $T^{(P)}$ and $P^{(P)}$. Then the matrix should be [$T^{(T)}$,$T^{(P)}$;$P^{(T)}$,$P^{(P)}$]. Note that to obtain $T^{(P)}$ and $P^{(P)}$, one has to impose a lower bound on the area fraction and this area fraction should be chosen so that $P^{(P)}$ is far from the Pareto front region of interest.
10	Number of points in first objective direction on hyperplane including the end points. If the Pareto front of three objectives are needed, the number of points in the other direction would be calculated by such that the spacings of the points are equal in each direction. The function that subdivides the hyperplane (`NNC_parameters.m`) cannot handle more than three objectives.

Table 4: Input arguments for `generate_pareto_front.m`.

Argument #	Description
11	Non-linear constraint types. Any combination of the following strings delimited by a comma: `Pmin` (lower bound on pressure drop) `Pmax` (upper bound on pressure drop), `Tmax` (upper bound on maximum temperature), `Amin` (lower bound on area fraction) `Amax` (upper bound on area fraction), `Vmin` (lower bound on volume fraction) `Vmax` (upper bound on volume fraction). Example: `'Pmax,Tmax'` or `'Tmax,Pmax'` (order immaterial)
12	Lower bound of pressure drop. Only applicable when #11 contains `'Pmin'`
13	Upper bound of pressure drop. Only applicable when #11 contains `'Pmax'`
14	Upper bound of maximum temperature. Only applicable when #11 contains `'Tmax'`
15	Lower bound of area/volume fraction. Only application when #11 contains `'Amin'`/`'Vmin'`
16	Upper bound of area/volume fraction. Only application when #11 contains `'Amax'`/`'Vmax'`
17	Logical: true (merge triangles used for geometrical constraints to form quadrilaterals)
18	'3D' conductivity
19	Domain thickness for calculating '2D' conductivity and volume fraction
20	Convection coefficient h_{conv} of $h_{conv}(T - T_{amb})$
21	Ambient temperature T_{amb} of $h_{conv}(T - T_{amb})$
22	Uniform heat source value. If analytical distributed heat source needs to be provided, use the two files `body_source_functions.cpp` in the directories mx_FEM and mx_sensitivity. The spatial derivative of the distributed heat source needs to be provided to the file located in mx_sensitivity. It is the responsibility of the user to ensure that the analytical forms in both files are consistent. In the mx_FEM directory, one must uncomment the directives `#define source_functions` in `compute_IGFEM_element.cpp` and `compute_regular_element.cpp`. Also, the same directive is to be uncommented in `IGFEM_element_pseudo_adjoint_forces.cpp`

3.3 Optimization of 'worst-case scenario'

Optimization of 'worst-case scenario' minimizes the worst objective function among a set of objective functions each obtained by blocking a subset of channels of the network. (The subset of blocked channels must not cause the solution to be undefined.) The minimax problem is converted into a single objective optimization following the formulation in (Charalambous and Conn [1]). The input arguments for `optimize_blocked_channels.m` are described in Table 10 and 6.

Table 5: Input arguments for `optimize_blocked_channels.m`.

Argument #	Description
1	Input channel file name (see User_manual in the IGFEM-Curves-2D directory about the channel input file format and the GUI tool used to generate the file)
2	File for imposing geometric constraints to prevent self-intersection (see Sec. 3.4). Set to [ ] if not applicable.
3	Either file with a set of initial designs (see Sec. 3.5) or input channel file if the initial designs are to be generated by choosing random values between the lower and upper bounds. Set to [ ] if the initial design is that given in the input channel file #1
4	File that list the channels to be blocked for determining the 'worst-case scenario' (See Sec. 3.6).
5	Either job number if #5 $>$ 0 or sample number if #5 $<$0 and #18 == 1. This argument only takes effect when an initial design file (#2) is given. If #5 $>$ 0, the sample number is calculated as (#5 - 1)$\times$#18 + current simulation number, where 1 $\leq$ current simulation number $\leq$ #18.
6	Output directory in the same directory as `main_blocked_channels.m`
7	File prefix for the output files
8	Vector of the number of elements in each direction
9	Vector of the domain bounds [xi,xf,yi,yf], where xi,xf,yi,yf are respectively the left, right, bottom and top coordinates of the domain
10	Cost function type. Only one type is available: `P_NORM` (p-norm of temperature field)
11	p of p-norm. Only applicable when #10==`P_NORM`
12	Non-linear constraint types. Any combination of the following strings delimited by a comma: `Pmin` (lower bound on pressure drop) `Pmax` (upper bound on pressure drop), `Tmax` (upper bound on maximum temperature), `Amin` (lower bound on area fraction) `Amax` (upper bound on area fraction), `Vmin` (lower bound on volume fraction) `Vmax` (upper bound on volume fraction). Example: `'Pmax,Tmax'` or `'Tmax,Pmax'` (order immaterial)

Table 6: Input arguments for `optimize_blocked_channels.m`.

Argument #	Description
13	Lower bound of pressure drop. Only applicable when #13 contains `'Pmin'`
14	Upper bound of pressure drop. Only applicable when #13 contains `'Pmax'`
15	Upper bound of maximum temperature. Only applicable when #13 contains `'Tmax'`
16	Lower bound of area/volume fraction. Only application when #13 contains `'Amin'`/`'Vmin'`
17	Upper bound of area/volume fraction. Only application when #13 contains `'Amax'`/`'Vmax'`
18	Number of simulations per job. For each job, #18 simulations will be run.
19	Logical: true (use the first initial design #3), false (use the network in #1)
20	Logical: true (merge triangles used for geometrical constraints to form quadrilaterals)
21	'3D' conductivity
22	Domain thickness for calculating '2D' conductivity and volume fraction
23	Convection coefficient h_{conv} of $h_{conv}(T - T_{amb})$
24	Ambient temperature T_{amb} of $h_{conv}(T - T_{amb})$
25	Uniform heat source value. If analytical distributed heat source needs to be provided, use the two files `body_source_functions.cpp` in the directories mx_FEM and mx_sensitivity. The spatial derivative of the distributed heat source needs to be provided to the file located in mx_sensitivity. It is the responsibility of the user to ensure that the analytical forms in both files are consistent. In the mx_FEM directory, one must uncomment the directives `#define source_functions` in `compute_IGFEM_element.cpp` and `compute_regular_element.cpp`. Also, the same directive is to be uncommented in `IGFEM_element_pseudo_adjoint_forces.cpp`

3.4 Geometrical constraint file

This file enables triangles/quads to be constructed for applying geometrical constraints. Note that only the use of triangles has been tested extensively and some data in the file is not applicable when quads are used. The format of the file is as follows:

```
polygons, <number of polygons,p>
```

```
v11 v12 ... v1n1
v21 v22 ... v2n2
...
vp1 vp2 ... vpnp
# where v11 v12 etc are vertex numbers

vertexCoords, <number of vertexCoords,c>
x11 x12
x21 x22
...
xc1 xc2

vertices2params, <number of maps, m>
vp11 vp12
vp21 vp22
...
vpm1 vpm2
# vpij means the design parameter corresponding the j coordinate of the
# ith vertex. If the vertex coordinate does not map to any design
# parameter, the corresponding entry should be nan

# the data from here onwards is only applicable when all polygons are
# triangles
restrictedParams, <number of restricted parameters, r>
val1
val2
...
valr
# restrictedParams are design parameters where the values are
# restricted to be equal to some other design parameters
# these extra parameters are introduced to increase the flexibility of
# the side triangles when applying the geometrical constraints
# val1, val2, ...are the inital values of these restricted parameters

paramPairs, r
pp11 pp12
pp21 pp22
...
ppr1 ppr2
# (pp11,pp12) is a pair consisting of a restricted parameter and
# a partner design parameter
```

```
sideTriangles, <number of side triangles, s>
triangle1
triangle2
...
trianglep
# these are the side triangles among those listed under polygons
```

Many examples can be found in the files ending with the extension ‘polygon’ in the directory ChannelFiles.

3.5 Initial design file

The initial design file (also called sample file) is used to start the optimization with different initial designs. The format of the file is as follows:

```
nSamples, <number of initial designs>
nParams, <number of design parameters>
val11
val12
...
val1nParams
...
...
valnSamples1
valnSamples2
...
valnSamplesnParams
```

Examples of the files can be found in the directory ChannelFiles with extensions ‘lhs’, ‘rand’, ‘rand1’ and ‘smmp’, which respectively indicate that the initial designs are generated using the Latin hypercube sampling method, shuffling within bounding boxes, random generation of quads for generating parallel networks and random generation of vertices followed by subgraph monomorphism to connect the vertices.

3.6 Blocked channel file

This file indicates which channels to block to produce a set of objectives corresponding to each blockages, among which the worst-case at each iteration is optimized for. One must ensure that none of the blockages produce an undefined objective.

```
<set 1 channel numbers>
<set 2 channel numbers>
...
<set n channel numbers>
```

Examples can be found in the directory blockedChannelFiles.

3.7 Running multiple jobs concurrently in LINUX

LINUX scripts are available to generate multiple main scripts so that multiple jobs can be run independently on different processors. Scripts for the SSM machines and the Taub cluster can be found in the directories SSM21_scripts and Taub_scripts. The scripts for the SSM machines can simply be run by issuing the command `./<script name with extension 'run'>` in the terminal. The scripts will generate a main MATLAB script for each job to be run and execute the jobs in the background. To run jobs in Taub, one must submit a pbs script by issuing the command `qsub <pbs script>`, which would then execute a script with the extension 'run'.

4 Outputs

Many files are produced at the beginning, during and after the optimization every iteration or every function evaluation. The files may be updated/replaced or a new one produced at each iteration. The description of the output files are shown in Table 7.

Table 7: Optimization output files.

Output file	Description
Simulation parameters	File with simulation parameters generated at beginning of optimization and located in the specified output directory. It also updated to indicate whether a simulation failed or succeed and record the duration of the simulation
Initial channel design	Channel file of starting design (very similar to channel file at iteration 0)
Channels	Channel file generated at each iteration starting with the specified file prefix and located in the specified output directory/sim⟨ simulation number⟩
Temperature	Vtk file of the temperature generated at each iteration starting with the specified file prefix and located in the specified output directory/sim⟨ simulation number⟩
Mesh and channels	MATLAB fig and jpg files of the mesh and the channels generated at each iteration starting with the specified file prefix and located in the specified output directory/sim⟨ simulation number⟩
Histories	Files with names that end with 'history' before the file extension. Various histories are saved as jpg files for real-time monitoring. The global variable `G_history` containing the histories of many variables of interest such as the objective values, pressure drop, maximum and average temperatures, area fraction etc is saved as the MATLAB file with name ending with 'history' before the extension 'mat'. These files are updated at every iteration.
High-frequency history	File with 'opt' extension recording the design parameter values, objective value and gradients at every function evaluation.
Pre- and post-optimization variables	MATLAB variable file that ends with 'allVariables' before the extension 'mat' containing the variables contained in the functions called by the main scripts. Saved at the beginning of a simulation and after the simulation is completed
Error files	Error files including the channel design just before the crash.

5 Examples

5.1 Example 1: Optimization of two-branch parallel network

Table 8: Input arguments for Sec. 5.1. Blank arguments are not applicable. SI units are used for all quantities except temperature, which is in °C.

Argument #	Value
1	'parallel2_start.channel'
2	'parallelTwo_NE.polygon'
3	'parallelTwo_NE_b.lhs'
4	-2
5	'parallel2'
6	'parallelTwo'
7	[30,40]
8	[0,0.15,0,0.2]
9	'P_NORM'
10	8
11	
12	
13	'Pmax'
14	
15	30000
16	
17	
18	
19	1
20	false
21	false
22	2.7
23	0.003
24	0
25	0
26	500

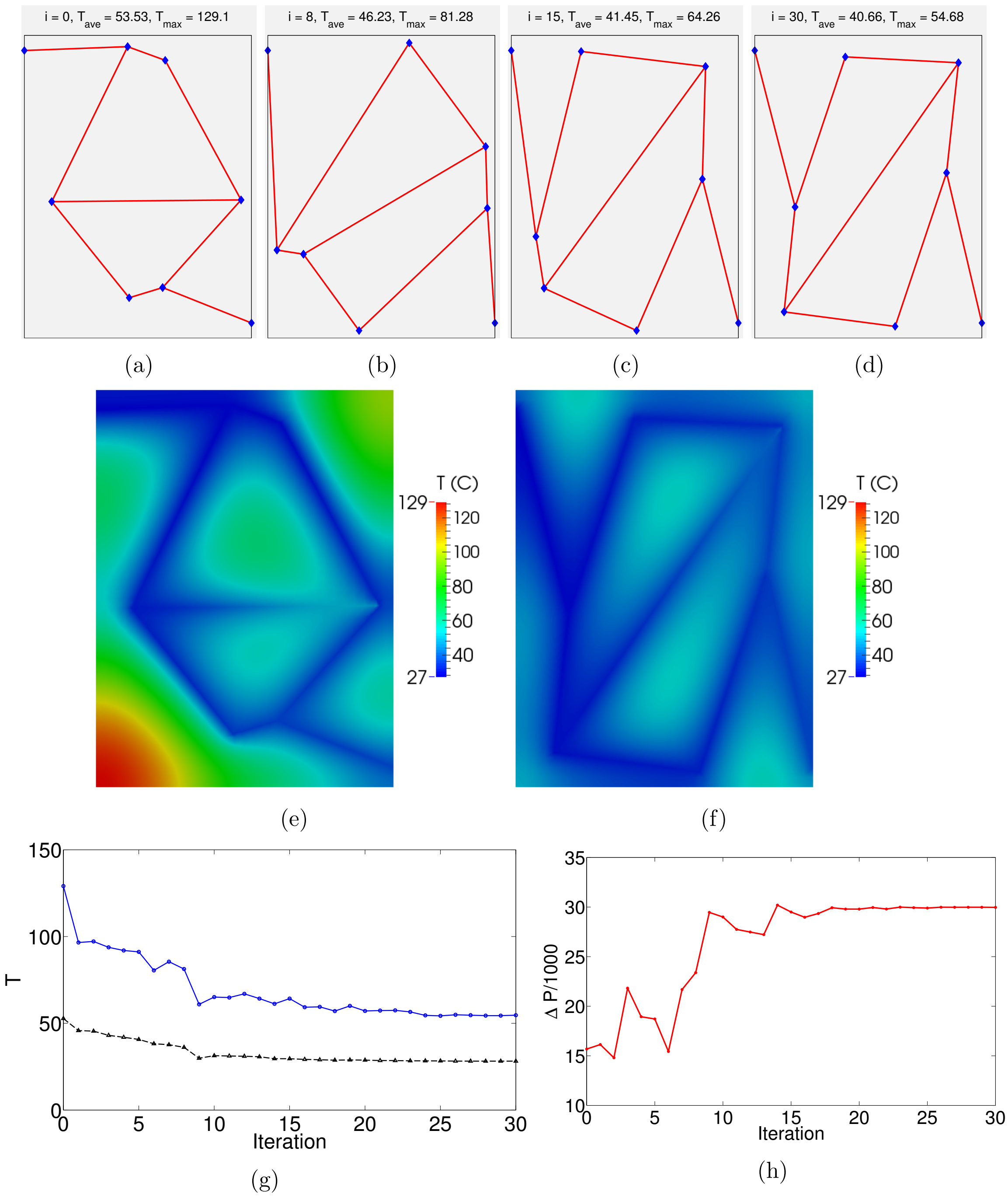

Figure 1: (a) - (d) Channel designs at iteration 0, 8, 15 and 30. (e), (f) The temperature distributions at iteration 0 and 30, obtained by plotting the generated vtk files in Paraview. (g) Histories of T_{max} (blue circles with solid line) and 8-norm of temperature (black triangles with dashed line), and (g) pressure drop.

5.2 Example 2: Generation of Pareto optimal front

Table 9: Input arguments for Sec. 5.2. Blank arguments are not applicable. SI units are used for all quantities except temperature, which is in °C.

Argument #	Value
1	‘parallelTwoNEoptimal.channel’
2	‘parallelTwo_NE.polygon’
3	‘parallel2Pareto’
4	‘parallelTwoPareto’
5	[30,40]
6	[0,0.15,0,0.2]
7	‘T,P’
8	8
9	[25.49,246.5; 46994,261]
10	60
11	[ ]
12	
13	
14	
15	
16	
17	false
18	2.7
19	0.003
20	0
21	0
22	500

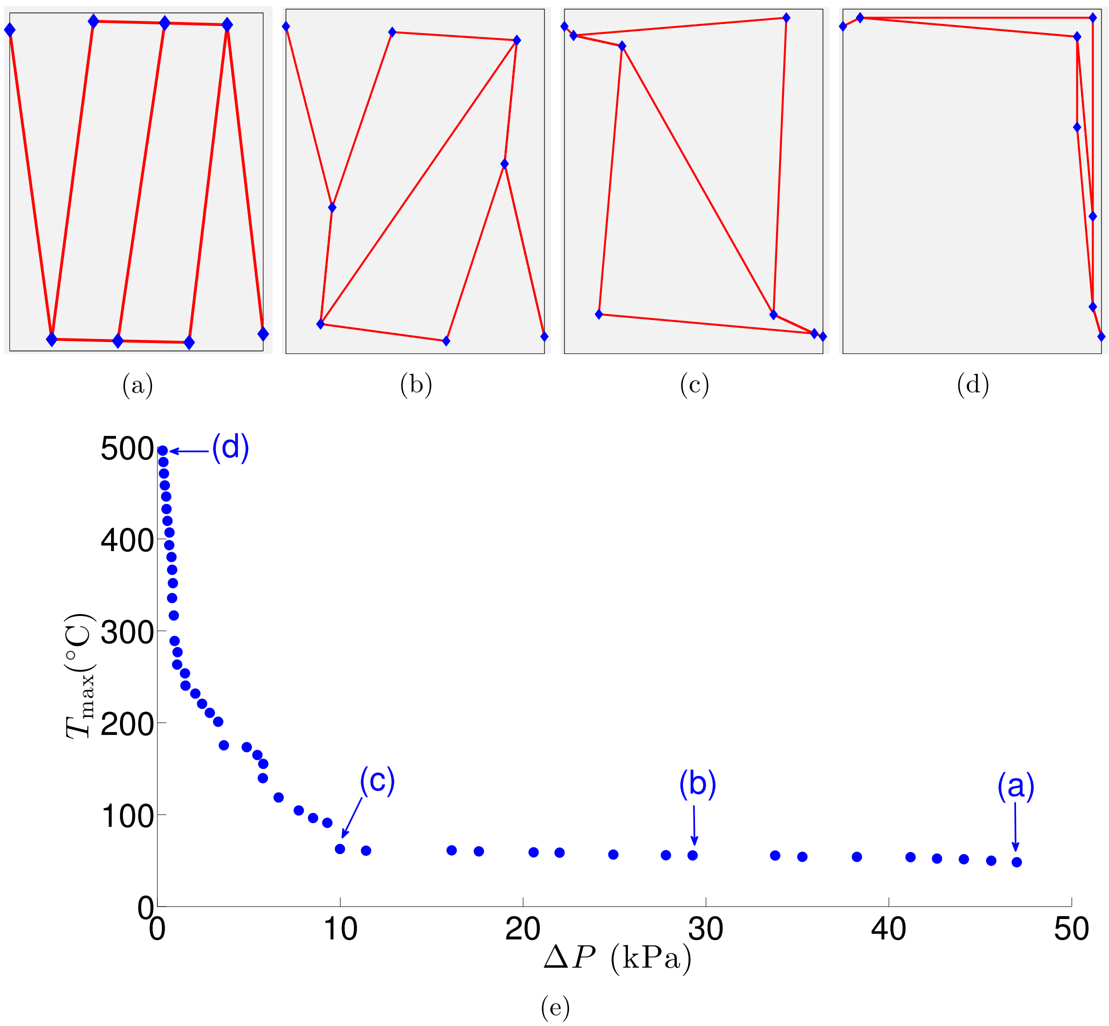

Figure 2: (a) Optimal design resulting from minimizing the 8-norm alone. (b), (c) Optimal designs along the Pareto front in (e).(d) Optimal design resulting from minimizing the pressure drop subject to a minimum area fraction of 0.018. (e) T_{max}-ΔP Pareto front obtained by postprocessing the 8-norm-ΔP Pareto points and filtering the resulting points using a Pareto filter.

5.3 Example 3: Optimization of 'worst-case scenario'

Table 10: Input arguments for `optimize_blocked_channels.m`.

Argument #	Description
1	‘parallel2x2ref.channel’
2	‘parallel2x2NE.polygon’
3	‘parallel2x2_1xp1_a.rand’
4	‘parallel2x2nBlk1.blk’
5	-22
6	‘parallel2x2’
7	‘insulated_composite’
8	[40,40]
9	[0,0.05,0,0.05]
10	’P_NORM’
11	8
12	[ ]
13	
14	
15	
16	
17	
18	1
19	false
20	false
21	2.7
22	0.003
23	0
24	0
25	500

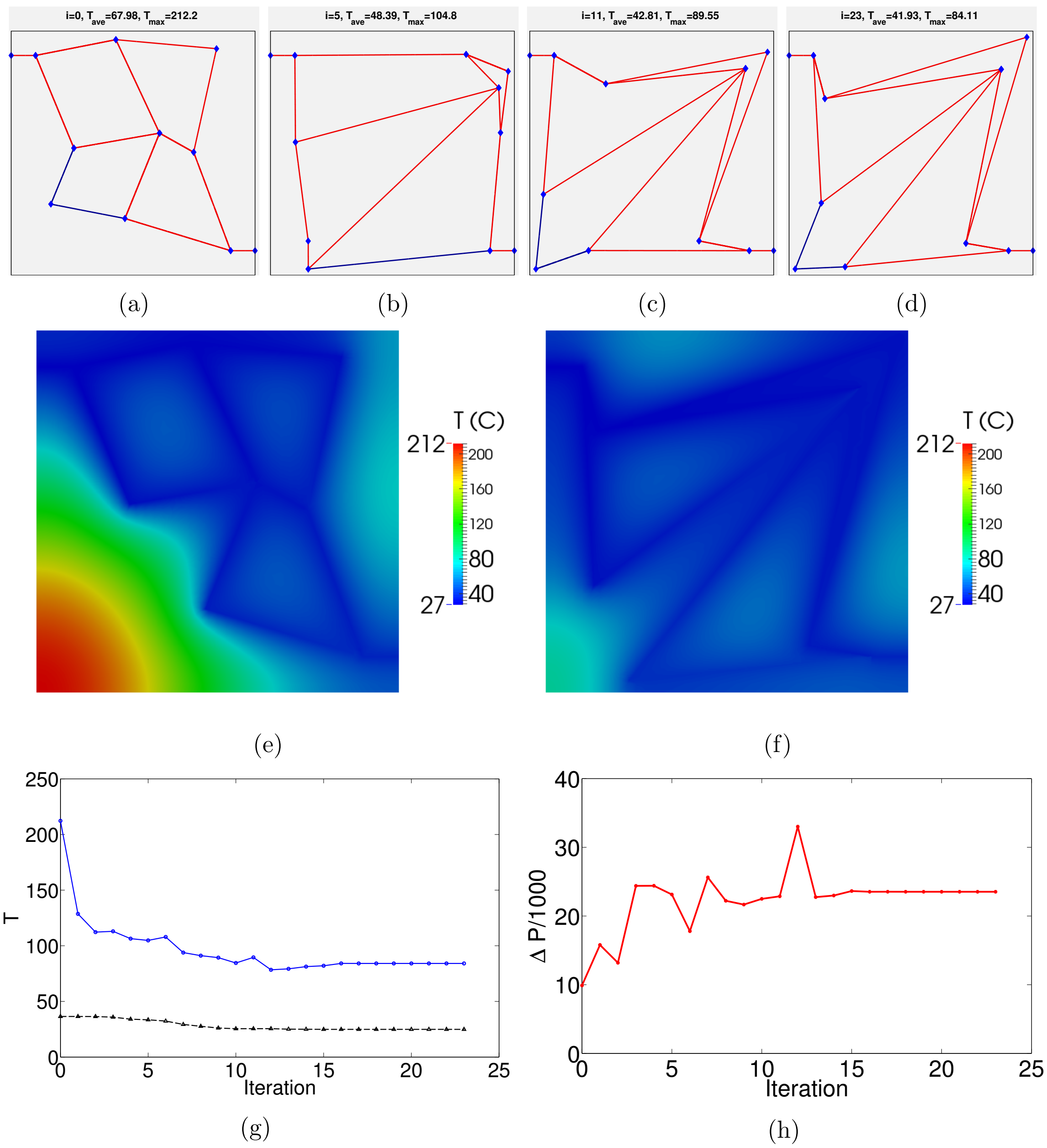

Figure 3: (a) - (d) Channel designs at iteration 0, 5, 11 and 23. (e), (f) The temperature distributions at iteration 0 and 30, obtained by plotting the generated vtk files in Paraview. (g) Histories of T_{max} (blue circles with solid line) and 8-norm of temperature (black triangles with dashed line), and (h) pressure drop.

6 References

References

[1] C. Charalambous and A. R. Conn. An efficient method to solve the minimax problem directly. *SIAM Journal on Numerical Analysis*, 15(1):162–187, 1978.

[2] A. Messac, A. Ismail-Yahaya, and C. A. Mattson. The normalized normal constraint method for generating the pareto frontier. *Struct Multidisc Optim*, 25:86–98, 2003.

[3] Ahmad R. Najafi, Masoud Safdari, Daniel A. Tortorelli, and Philippe H. Geubelle. A gradient-based shape optimization scheme using an interface-enriched generalized FEM. *Comput. Methods Appl. Mech. Eng.*, 296:1–17, 2015.

[4] Marcus Hwai Yik Tan, Ahmad R. Najafi, Stephen J. Pety, Scott R. White, and Philippe H. Geubelle. Gradient-based design of actively-cooled microvascular composite panels. *In preparation for Int J Heat Mass Transfer*, 2016.