JPL

Ice Sheet System model
Software Architecture

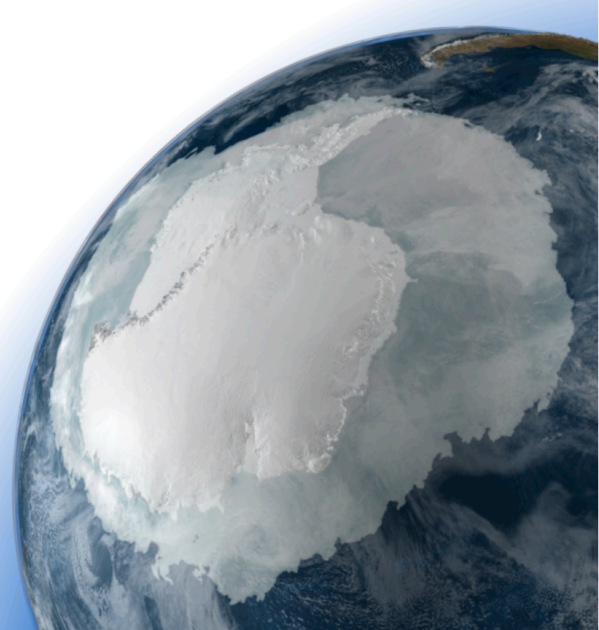
Eric LAROUR[1], Eric RIGNOT[1,3], Mathieu MORLIGHEM[1,2], Hélène SEROUSSI[1,2] Chris BORSTAD[1], Feras HABBAL[1,3], Daria HALKIDES[1,4], Behnaz KHAKBAZ[1], John SCHIERMEIER[1], Nicole SCHLEGEL[1]

[1]Jet Propulsion Laboratory - California Institute of Technology

[2]Laboratoire MSSMat, École Centrale Paris, France

[3]University of California, Irvine

[4]Joint Institute for Regional Earth System Science & Engineering, UCLA



ISSM

Outline

Introduction

- ISSM is written in C++ (C-style)
- ISSM is hosted in MATLAB using the mex API
- ISSM is written to run serial/parallel seamlessly
- ISSM is configured to run using the autotools suite for multi-platform support

Introduction

- ISSM is written in C++ (C-style)
- ISSM is hosted in MATLAB using the mex API
- ISSM is written to run serial/parallel seamlessly
- ISSM is configured to run using the autotools suite for multi-platform support

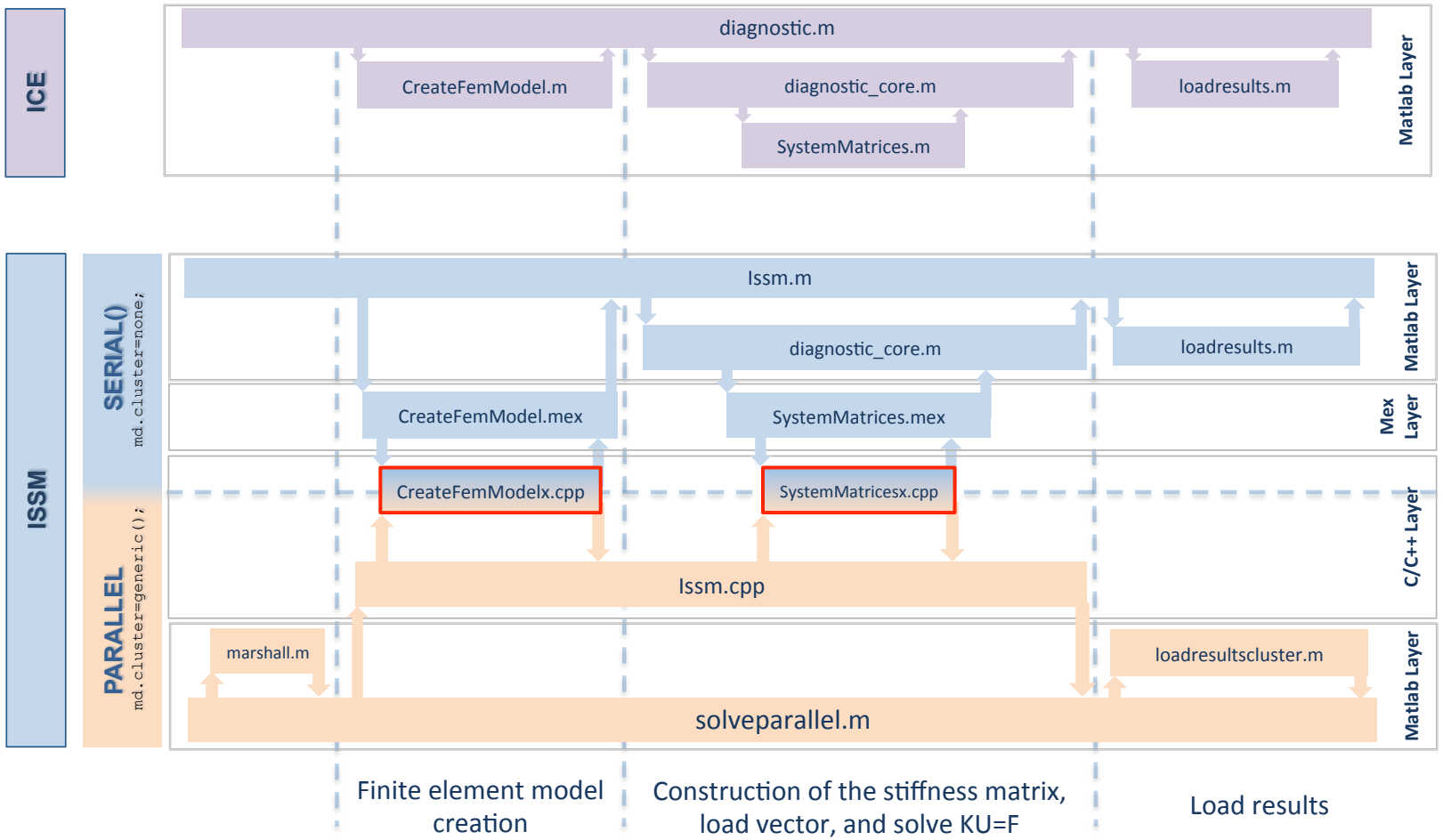
Introduction

- ISSM is written in C++ (C-style)
- ISSM is hosted in MATLAB using the mex API
- ISSM is written to run serial/parallel seamlessly
- ISSM is configured to run using the autotools suite for multi-platform support

Introduction

- ISSM is written in C++ (C-style)
- ISSM is hosted in MATLAB using the mex API
- ISSM is written to run serial/parallel seamlessly
- ISSM is configured to run using the autotools suite for multi-platform support

General Layout



General Layout

```
$ pwd
/Users/cborstad/issmuci/trunk
$ ls
AUTHORS      NEWS           configs       examples          missing
COPYING      README         configure.ac  execution         scripts
ChangeLog    bin            cron          externalpackages  src
INSTALL      compile        depcomp       install-sh        startup.m
LICENSE      config.guess   doc           lib               startup.py
Makefile.am  config.sub     etc           m4                test
```

General Layout

The core of the code is in trunk/src, for C++, mex API, MATLAB routines, etc.

```
$ pwd
/Users/cborstad/issmuci/trunk/src
$ ls
Makefile.am  ad  c  dox  m  mex  perl  pro  py
```

The core of the C/C++ code is in trunk/src/c

```
$ pwd
/Users/cborstad/issmuci/trunk/src/c
$ ls
Container         include            issm.h   shared     toolkits
EnumDefinitions   intel-compile.sh   modules  solutions
Makefile.am       io                 objects  solvers
```

C/C++ Code

Solutions are high-level drivers of C/C++ routines:

```
$ pwd
/Users/cborstad/issmuci/trunk/src/c/solutions
$ ls
AdjointCorePointerFromSolutionEnum.cpp  diagnostic_core.cpp
AnalysisConfiguration.cpp               enthalpy_core.cpp
CorePointerFromSolutionEnum.cpp         gradient_core.cpp
ProcessArguments.cpp                    hydrology_core.cpp
ResetBoundaryConditions.cpp             hydrology_core_step.cpp
WriteLockFile.cpp                       issm.cpp
adjointbalancethickness_core.cpp        objectivefunctionC.cpp
adjointdiagnostic_core.cpp              prognostic_core.cpp
balancethickness_core.cpp               solutions.h
bedslope_core.cpp                       steadystate_core.cpp
control_core.cpp                        steadystateconvergence.cpp
controlconvergence.cpp                  surfaceslope_core.cpp
controlrestart.cpp                      thermal_core.cpp
controltao_core.cpp                     thermal_core_step.cpp
convergence.cpp                         transient_core.cpp
```

C/C++ Code

Solutions call other modules or solutions. Femmodel is driving class.

```
$ pwd
/Users/cborstad/issmuci/trunk/src/c/solutions
$ cat surfaceslope_core.cpp
/*!\file: surfaceslope_core.cpp
...
#include "./solutions.h"
...
void surfaceslope_core(FemModel* femmodel){
/*parameters: */
int dim;
...
/*Recover some parameters: */
femmodel->parameters->FindParam(&dim,MeshDimensionEnum);
...
/*Call on core computations: */
...
solver_linear(femmodel);
...
if(solution_type==SurfaceSlopeSolutionEnum && !control_analysis){
InputToResultx(femmodel->elements,femmodel->nodes,femmodel->vertices,
femmodel->loads,femmodel->materials,femmodel->parameters,SurfaceSlopeXEnu
}
...
}
```

C/C++ Code

Extensive list of modules that drive the computations:

```
$ pwd
/Users/cborstad/issmuci/trunk/src/c/modules
$ ls
```

AddExternalResultx	InputUpdateFromConstantx	OutputRiftsx
AverageFilterx	InputUpdateFromDakotax	ParsePetscOptionsx
AverageOntoPartitionx	InputUpdateFromMatrixDakotax	PointCloudFindNeighborsx
BamgConvertMeshx	InputUpdateFromSolutionx	PropagateFlagsFromConnectivityx
BamgTriangulatex	InputUpdateFromVectorDakotax	Reduceloadx
Bamgx	InputUpdateFromVectorx	Reducevectorgtofx
Chacox	InterpFromGridToMeshx	Reducevectorgtosx
ComputeBasalStressx	InterpFromMesh2dx	RequestedOutputsx
ComputeStrainRatex	InterpFromMeshToGridx	ResetConstraintsx
ConfigureObjectsx	InterpFromMeshToMesh2dx	ResetCoordinateSystemx
ConstraintsStatex	InterpFromMeshToMesh3dx	Responsex
ContourToMeshx	IoModelToConstraintsx	RheologyBbarAbsGradientx
ContourToNodesx	KMLMeshWritex	Scotchx
ControlInputGetGradientx	KMLOverlayx	Shp2Kmlx
ControlInputScaleGradientx	Kml2Expx	Solverx
ControlInputSetGradientx	Ll2xyx	SpcNodesx
CostFunctionx	MassFluxx	StringToEnumx
CreateNodalConstraintsx	MaxAbsVxx	SurfaceAbsVelMisfitx
DakotaResponsesx	MaxAbsVyx	SurfaceAreax
Dakotax	MaxAbsVzx	SurfaceAverageVelMisfitx
DragCoefficientAbsGradientx	MaxVelx	SurfaceLogVelMisfitx
ElementConnectivityx	MaxVxx	SurfaceLogVxVyMisfitx
ElementResponsex	MaxVyx	SurfaceRelVelMisfitx
EnumToStringx	MaxVzx	SystemMatricesx
Exp2Kmlx	Mergesolutionfromftogx	ThicknessAbsGradientx
GetSolutionFromInputsx	MeshPartitionx	ThicknessAbsMisfitx
GetVectorFromInputsx	MeshProfileIntersectionx	TimeAdaptx
Gradjx	MinVelx	TriaSearchx
GroundinglineMigrationx	MinVxx	UpdateConstraintsx
HoleFillerx	MinVyx	UpdateDynamicConstraintsx
IceVolumex	MinVzx	UpdateVertexPositionsx
InputArtificialNoisex	ModelProcessorx	VecMergex
InputControlUpdatex	NodalValuex	VerticesDofx
InputConvergencex	NodeConnectivityx	Xy2llx
InputDuplicatex	NodesDofx	modules.h
InputScalex	Orthx	

C/C++ Code

Typical module to create stiffness and loads:

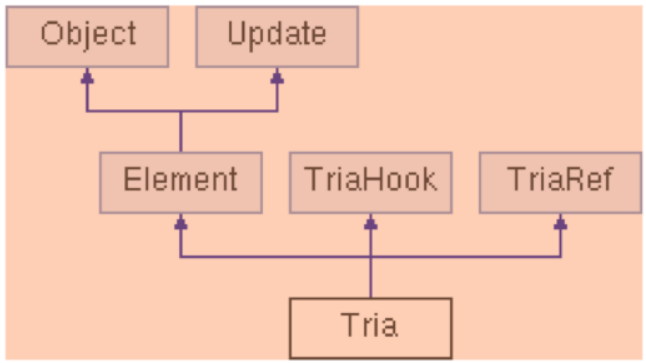
```
$ pwd
/Users/cborstad/issmuci/trunk/src/c/modules/SystemMatricesx
$ ls
SystemMatricesx.cpp  SystemMatricesx.h
```

```
...
   if(kflag){

Kff=NewMat(fsize,fsize,connectivity,numberofdofspernode);
Kfs=NewMat(fsize,ssize,connectivity,numberofdofspernode);
df=NewVec(fsize);

/*Fill stiffness matrix from elements: */
for (i=0;i<elements->Size();i++){
element=(Element*)elements->GetObjectByOffset(i);
element->CreateKMatrix(Kff,Kfs,df);
}

/*Fill stiffness matrix from loads if loads have the current
for (i=0;i<loads->Size();i++){
load=(Load*)loads->GetObjectByOffset(i);
if (load->InAnalysis(configuration_type)) load->CreateKMatrix(Kff,Kfs);
}
...
```



MATLAB

MATLAB routines for pre-post processing:

```
$ pwd
/Users/cborstad/issmuci/trunk/src/m
$ ls
classes  enum  kml  model  planet  qmu  shared  solutions  solvers  utils
$ pwd
/Users/cborstad/issmuci/trunk/src/m/model
$ ls
BasinConstrain.m                graddetection.m                  process_solve_options.m
BasinConstrainShelf.m           ismodelselfconsistent.m          processgeometry.m
DepthAverage.m                  kmlimagesc.m                     project2d.m
MeltingGroundingLines.m         loadmultipleresultsfromcluster.m project3d.m
PropagateFlagsUntilDistance.m   loadresultsfromcluster.m         qpr.m
README                          loadresultsfromdisk.m            qstat.m
SectionValues.m                 marshall.m                       radarpower.m
ThicknessCorrection.m           mechanicalproperties.m           recover_areas.m
WriteData.m                     mesh                             regionaltransient2d.m
addnote.m                       misfit.m                         setflowequation.m
averageconnectivity.m           modelextract.m                   setmask.m
averaging.m                     modelsextract.m                  setmask2.m
basalstress.m                   modelsextractfromdomains.m       shear2d.m
basevert.m                      modis.m                          sia.m
bedslope.m                      multiplequeue                    slope.m
collapse.m                      outflow.m                        solve.m
contourenvelope.m               parametercontroloptimization.m   solveparallel.m
contourmassbalance.m            parameterization                 solvers
display                         parseresultsfromdisk.m           thicknessevolution.m
divergence.m                    partition                        tres.m
drivingstress.m                 petscversion.m                   waitonlock.m
effectivepressure.m             plot
extrude.m                       printmodel.m
```

MATLAB

Augmented by mex routine:

```
$ pwd
/Users/cborstad/issmuci/trunk/src/mex
$ ls
AddExternalResult           InputConvergence          ParsePetscOptions
AverageFilter               InputDuplicate            PointCloudFindNeighbors
BamgConvertMesh             InputScale                ProcessParams
BamgMesher                  InputToResult             PropagateFlagsFromConnectivity
BamgTriangulate             InputUpdateFromConstant   Reduceload
Chaco                       InputUpdateFromDakota     Reducevectorgtof
ComputeBasalStress          InputUpdateFromSolution   Reducevectorgtos
ConfigureObjects            InputUpdateFromVector     ResetCoordinateSystem
ConstraintsState            InternalFront             Response
ContourToMesh               InterpFromGridToMesh      Scotch
ContourToNodes              InterpFromMesh2d          Shp2Kml
ControlInputGetGradient     InterpFromMeshToGrid      Solver
ControlInputScaleGradient   InterpFromMeshToMesh2d    SparseToVector
ControlInputSetGradient     InterpFromMeshToMesh3d    SpcNodes
ControlOptimization         KMLMeshWrite              StringToEnum
CostFunction                KMLOverlay                SurfaceArea
CreateNodalConstraints      Kml2Exp                   SystemMatrices
Dakota                      Ll2xy                     Test
DakotaResponses             Makefile.am               TimeAdapt
Echo                        Mergesolutionfromftog     TriMesh
ElementConnectivity         MeshPartition             TriMeshNoDensity
EnumToString                MeshProfileIntersection   TriMeshProcessRifts
Exp2Kml                     ModelProcessor            TriMeshRefine
GetSolutionFromInputs       NodeConnectivity          TriaSearch
GetVectorFromInputs         NodesDof                  UpdateConstraints
Gradj                       Orth                      UpdateDynamicConstraints
GroundinglineMigration      OutputResults             UpdateVertexPositions
HoleFiller                  OutputRifts               VerticesDof
InputControlUpdate          ParameterOutput           Xy2ll
```

MATLAB

Typical mex routine:

```
$ pwd
/Users/cborstad/issmuci/trunk/src/mex/SystemMatrices
$ ls
SystemMatrices.cpp  SystemMatrices.h
void mexFunction( int nlhs, mxArray* plhs[], int nrhs, const mxArray* prhs[]){

/*input datasets: */
Elements   *elements   = NULL;
...
/* output datasets: */
Mat    Kff  = NULL;
...
/*Boot module: */
MODULEBOOT();
...
/*Input datasets: */
FetchMatlabData((DataSet**)&elements,ELEMENTS);
...
SystemMatricesx(&Kff,&Kfs,&pf,&df,&kmax,elements,nodes,vertices,loads,materials,parameters,kflag,pflag
...
/*write output datasets: */
WriteMatlabData(KFF,Kff);
...
/*end module: */
MODULEEND();
```

Conclusions

- ISSM is at its core a C++ language, wrapped into Matlab
- ISSM can run both in serial and parallel mode, either by relying on mex modules calling the C-code, or on issm.exe compiled out of the entire C++ code
- More features are available, such as:
  - Nighlty runs and test suites
  - Autotools to automatically install on any platform
  - Automatic documentation


Thanks!