

The Open Source model framework ENKI

What is ENKI?

ENKI is a modular framework for implementing hydrological or other environmental models. Both lumped and distributed models are supported. ENKI builds a model from a set of user-defined subroutines, which operate on GIS data within a spatial region.

Also providing calibration and evaluation functionality, ENKI makes it easy for model developers to implement and test single routines and various model compositions in a fixed framework. ENKI is now released as open source under GNU LGPL.

SnowReflect	Hydra	Soil	HBVResponse	Qsubcat	SimpleR	esponse	SimplQsubcat
Consthum	Constwind	Idwrad	BayesTkrig	PcorrMap2	HydraEP	HydraCanopy	GamSno
LocalName	Usage	DataType	Connection	Description			
StatElev TargetElev Nugget Sill Range Zscale PriEtgrad PriSDtgrad StatTemp TargetTemp EmpEstTgrad PostEtgrad	static static parameter parameter parameter parameter parameter input response response response	network raster scalar scalar scalar scalar scalar network raster scalar scalar scalar	tstats_elev Elevation Tnugget Tsill Trange Tzscale (routine local) PriSDtgrad tstats GridTemp EmpEstTgrad PostEtgrad	XY distance with Prior expectation Prior standard de Measured statio Interpolated air t Frequentistic est	et locations [ma ariogram [unit^2 semivariogram] ential semivariog a same semivar of lapse rate [o ev of lapse rate [o n temperature] emperature [de timate of lapse	ap units] 2] [unit^2] gram [map units] as one vertical uni degC/100m] [degC/100m] [degC]	t [-]
-Interpolated ai	r temperature (degC]					
TargetTemp			O So	calar (© Raster	○ Network	GridTemp	

The ENKI framework recognises the number, types, and names of each subroutine variable. The framework then exposes the variables to the user within the proper context, ensuring that:

- The model is completely and consistently set up
- Distributed maps coincide spatially where necessary
- Time series exist for input variables
- State variables are initialised for the correct date/time
- GIS data sets exist for static map data

Comparison	Test data	Reference data	Start Time	End Time	weights
Temporal R2	SimDischarge	OBSdischarge	01.09.2000	31.08.2001	Equal
Temporal R2	SimplSimDischarge	OBSdischarge	01.09.2000	31.08.2001	Equal
Temporal Likelihood	SimplSimDischarge	OBSdischarge	01.09.2000	31.08.2001	Equal
Spatial Likelihood	NDSI_SIM	NDSI_OBS	01.09.2000	31.08.2001	Equal
Temporal AR(1) Likelihood	SimplSimDischarge	OBSdischarge	01.09.2000	31.08.2001	Equal
		w Output / PM spe		Simulated output va	ariable Referen
				Simulated output va	ariable Referen
		omparison type		NDSI_SIM	▼ NDSI_
		omparison type Spatial StdErr Spatial R2 Spatial R2 In			▼ NDSI_
		omparison type patial StdErr patial R2 patial R2 In patial StdErr	₹ I r between two	NDSI_SIM	▼ NDSI_
		omparison type patial StdErr patial R2 patial R2 In patial StdErr patial SumDev emporal AR(1) Likeli	▼ I r between two	NDSI_SIM	▼ NDSI_
2000		omparison type Spatial StdErr Spatial R2 Spatial R2 In Spatial StdErr Spatial SumDev Semporal AR(1) Likeli Semporal AvgDev	▼ [I ► r between two	NDSI_SIM o distributed data si Start	NDSI
New Edit D	elete	omparison type patial StdErr patial R2 patial R2 In patial StdErr patial SumDev emporal AR(1) Likeli	▼ [I ► r between two	NDSI_SIM	▼ NDSI_

Contacts:

Sjur Kolberg, SINTEF sjur.kolberg@sintef.no Oddbjørn Bruland, Statkraft oddbjorn.bruland@statkraft.nc





Who is ENKI for?

ENKI offers functionality for three different levels of involvement in model construction:

Model application:

Run and evaluate pre-built models for any response using several objective functions, choose search algorithm for calibration, and analyse uncertainty arising from input errors or parameter equifinality.

Model analysis: Add or replace subroutines, run multi-model ensembles, switch between calibrated and mapped parameters, and experiment with different distribution schemes without having to write or compile source code.

Routine implementation and testing: Code the core of a new lumped or distributed subroutine, include it in an ENKI model, and let ENKI handle all model administration and interface code.

As ENKI continues to develop as a experimental tool, its core API is also being implemented in Statkraft's forecast system for operational hydropower. This common core, the modular design, and the open source license all facilitate rapid dissemination of new methods into operational use.

Distribution	Value	Variance	Min	Max		
Uniform 📃	2	12	1	12	Set	
Constant	, rent value: 8.34438)	1	1			
Uniform	Territ value. 0.04400)					MC method
Normal	Routine	Minimum	Maximum	Distribution		
Gamma	HydraSoil	0	3.40282E+	222.5		Marquardt-Levenberg
Beta	HydraSoil	0	3.40282E+	100		Multi-surface gradient
etmp	HydraEP	-3.40282E+	3.40282E+	0.16		search using the Jacobian
consthum	Consthum	-3.40282E+	3.40282E+	80		matrix (PEST algorithm)
BETA	HydraSoil	-3.40282E+	3.40282E+	Uniform(0.5,5)		
ResetSnowDepth	GamSnow	0	3.40282E+	30		O SCE-UA
tysum	HydraEP	0	3.40282E+	Uniform(100,1000)		Global shuffled complex
Constwind	Constwind	-3.40282E+	3.40282E+	1	=	evolution. Slow and robust
TX	PcorrMap2;GamS	-3.40282E+	3.40282E+	Uniform(-3,3)		for difficult cases.
Rtreshold	HBVResponse	-3.40282E+	3.40282E+	48		
LP	HydraSoil	-3.40282E+	3.40282E+	0.9		C Random MC (GLUE)
RadGrad	Idwrad	-3.40282E+	3.40282E+	0		Random sampling from
FastDecayRate	GamSnow	0	3.40282E+	Uniform(1,12)		specified distributions
Windconst	GamSnow	-3.40282E+	3.40282E+	Uniform(0.5,9)		
MaxIntDist	Idwrad	0	3.40282E+	300000		DREAM MCMC
Tsill	BayesTkrig	0	3.40282E+	6		Adaptive Metropolis
esnw	HydraEP	-3.40282E+	3.40282E+	0.1		sampler, requires
epcorr	HydraEP	-3.40282E+	3.40282E+	1.2		likelihood-based PMs
Maxalbedo	GamSnow	0	1	0.9		
MaxLWC	GamSnow	0	1	0.1		C Conditional Univariate
Trange	BayesTkrig	0	3.40282E+	50000		Univariate profiling around
Tzscale	BayesTkrig	0	3.40282E+	20		the current location
ewnd	HydraEP	-3.40282E+	3.40282E+	0.6		
perc	HB∨Response		3.40282E+			C External list
k0	HB∨Response	-3.40282E+	3.40282E+	0.007		Parameter sets read from
MaxIntStats	Idwrad	0	3.40282E+	25		pre-existing file
SurfaceLaver	GamSnow	0	3 40282E+	50	.	
	Set Seed #	MC runs:	🔲 Store o	utput		
Set file	0		Set PM w	veights	Can	cel OK

egModel: EnkiSimpleTest - SentralReg	Date Tests Test Mind	the theory		
View Model Region Parameters Initial	s Input Outputs <u>H</u> elp			
≇∎IX ®n @n @n @n ? ₩? model			Region: SentralReg	
art date 01.08.2000	Set parameters Set initials	Pause	Raster GRIB_longrad File set: GRIB_longrad Boundaries = region's default Dimensions = 200 rows * 130 Missing-code = -99.000000	columns
d date 31.08.2001 rrent 01.08.2000	Set outputs Setup MC autocal		A0 A1 A2 AccMeltDepth	New scalar New raster
active			acctemp actevaph Albedo AverageRunoff BETA	New network Delete
Single run MC param run Forecast ru	n Sim forecasts Ensemb	le run Sim ensemble	Catchments CONIFER_FRAC consthum Constwind	Input Database Output Database
	New databaseImport ASCII tableImport raster groupImport GRIB dataImport radar dataImport ICC data (test)	Attributes Attributes Attributes Attributes Attributes Ntw OBSdischarge(2710,1,1, Ntw rstats(2710,1,1,1,23) Rst GridPrec(2710,1,1,23) Rst OBSSCA(57,1,1,1,200,13) Rst NDSI_OBS(83,1,1,1,200,	EmpEstTgrad epcorr eprc	Metadata Statistics
	Export ASCII table Export raster group Close		GLACIERS GRIB_geopot GRIB_longrad GRIB_prec GRIB_precdur GRIB_rhum GRIB_shortrad	Set Files Read data Write data
y				NUM

Why distributed models?

Mountainous areas exhibit strong gradients in meteorology, topography and land surface properties.

For nonlinear processes, the use of catchment averages in model equations lead to biased results. For catchment sizes of 10²-10³ km², model errors depend more on heterogeneity and uncertainty in input data, than on inadequate model equations. Spatial distribution allows different response from various parts of the catchments, and emphasises interpolation and downscaling of input data.

Why a regional calibration approach?

Parameter equifinality and poor runoff data in regulated basins encourage the use of several series to reduce the information deficit.

A set of gauged basins is seen as a sample representing the region, enabling estimation of uncertainty also for the ungauged parts.

In Norwegian mountains, regional calibration reduces Nash-Sutcliffe values by 0.05-0.07 compared to catchment specific calibration. Sensitivity analyses emphasise the meteorology-related parameters as the most important.

Operationally, it is easier to maintain a common model for several sub-basins, than to calibrate, feed and update a model for each reservoir. Forecasts are increasingly needed for arbitrary spatial domains; stream intakes, electricity market regions, or river sections with legal flow requirements.



State of software

A minimal ENKI subroutine

ENKI is now developing both as a research tool and as a simulation engine for an operational forecast system. Currently it builds under Windows and Visual Studio; efforts to remove these platform and compiler dependency has started.

Recent modifications include a full separation of API and user interface, making it possible to run ENKI from GIS programs and other software environments.

Source code and binaries are available from the authors, released under the GNU LPGL license.

Technology

ENKI is written in C++, and uses a plug-in structure to invoke the subroutines. These are built separately as dynamic-link libraries (DLLs). All subroutines are coded as sub-classes of a generic method class, which is known by the ENKI framework. The subroutine programmer can rely on a few routines being called in specific situations:

- The constructor is called when the user includes the method in a model, and informs the ENKI framework about the routine's variable interface.
- Init() is called when the model is linked to a specific region, and all the routine's variables are linked to GIS data objects with known spatial extent. Optional.
- **PreProcess()** is called when all parameter values are set, thus for each iteration during auto-calibration. Optional.
- **Respond**() is called for each time step, and implement the process equations.
- **Calc()** may replace Respond() when the routine is purely vertical.

Vertical routines implementing Calc() rely on the framework for spatial looping, and can be used in lumped or distributed models without adaptation. Other routines may combine variables with different geometry.

The source code of an ENKI routine is highly compact. Below is the source code for a linear reservoir, admittedly with the simplest possible numeric solution. LinearTank::LinearTank() // Constructor, defining the variable interface outflow = newmethvar("outflow",true,"methvar","response","Outflow from linear tank"); storconst = newmethvar("storconst", true,"methvar","parameter","Response tank time constant",0,1) inflow = newmethvar("inflow", true,"methvar","input","Inflow to linear tank"); storage = newmethvar("storage", true,"methvar","state","Response tank storage in mm",0); bool LinearTank::Calc() // Response function, representing a single linear tank storage->m_value += inflow->m_value; outflow->m_value = storage->m_value * storconst->m_value; storage->m_value -= outflow->m_value; return true;