A Multi Fidelity Ensemble Kalman Filter with a machine learned surrogate model

Jeffrey van der Voort

joint work with: Martin Verlaan Hanne Kekkonen

Delft University of Technology, The Netherlands

June 18th, 2024

Applications

Use of ensemble simulations and ensemble data assimilation is ubiquitous in geosciences...

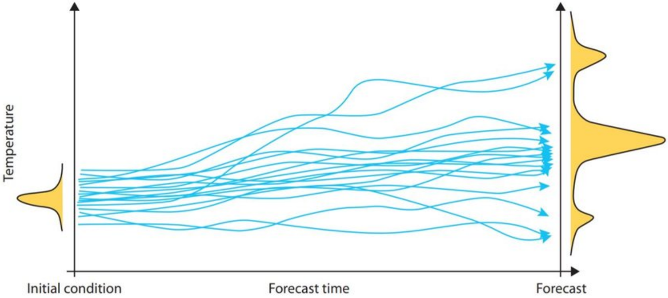


Figure: Figure 1 from Grönquist et al. (2019): *Predicting Weather Uncertainty with Deep Convnets*

Applications

Use of ensemble simulations and ensemble data assimilation is ubiquitous in geosciences...

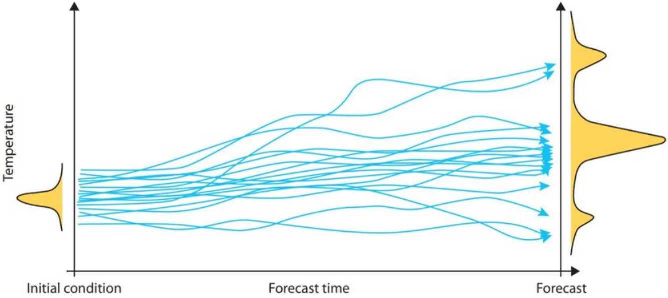


Figure: Figure 1 from Grönquist et al. (2019): *Predicting Weather Uncertainty with Deep Convnets*

...but running ensembles can be very expensive.

Main idea

Main idea

Few expensive, but accurate, full order model runs

Main idea

Few expensive, but accurate, full order model runs

+

Many cheap, but less accurate, surrogate model runs

Main idea

Few expensive, but accurate, full order model runs

$+$

Many cheap, but less accurate, surrogate model runs

$\Rightarrow$ combined using a Multi Fidelity Ensemble Kalman Filter (MF-EnKF) framework.

The Multi-Fidelity EnKF (MF-EnKF)

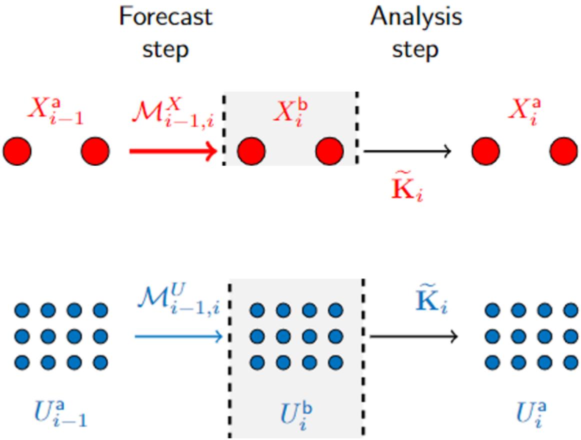
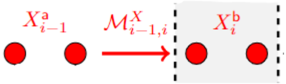
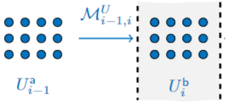


Figure: Figure 4.1 [edited] from Popov et al. (2021): *A Multifidelity Ensemble Kalman Filter with Reduced Order Control Variates*

The Multi-Fidelity EnKF (MF-EnKF) - forecast step

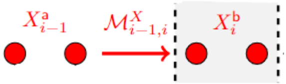


$N_X = 2$

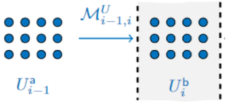


$N_U = 12$

The Multi-Fidelity EnKF (MF-EnKF) - forecast step



$N_X = 2$



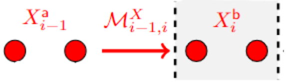
$N_U = 12$

Forecast step

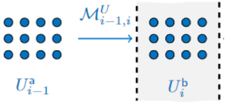
$$X_i^{b,[k]} = \mathcal{M}^X(X_{i-1}^{a,[k]}) + \varepsilon_i^X, \qquad k = 1, \ldots N_X$$

$$U_i^{b,[k]} = \mathcal{M}^U(U_{i-1}^{a,[k]}) + \varepsilon_i^U, \qquad k = 1, \ldots N_U$$

The Multi-Fidelity EnKF (MF-EnKF) - forecast step



$N_X = 2$



$N_U = 12$

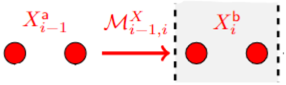
Forecast step

$$X_i^{b,[k]} = \mathcal{M}^X(X_{i-1}^{a,[k]}) + \varepsilon_i^X, \qquad k = 1, \ldots N_X$$

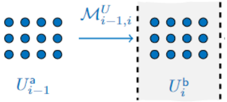
$$U_i^{b,[k]} = \mathcal{M}^U(U_{i-1}^{a,[k]}) + \varepsilon_i^U, \qquad k = 1, \ldots N_U$$

- $\mathcal{M}^X$ propagates state forward from time t_{i-1} to t_i

The Multi-Fidelity EnKF (MF-EnKF) - forecast step



$N_X = 2$



$N_U = 12$

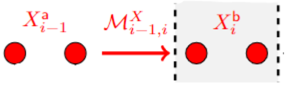
Forecast step

$$X_i^{b,[k]} = \mathcal{M}^X(X_{i-1}^{a,[k]}) + \varepsilon_i^X, \qquad k = 1, \dots N_X$$

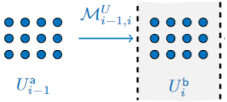
$$U_i^{b,[k]} = \mathcal{M}^U(U_{i-1}^{a,[k]}) + \varepsilon_i^U, \qquad k = 1, \dots N_U$$

- $\mathcal{M}^X$ propagates state forward from time t_{i-1} to t_i
- $\mathcal{M}^U$ is a machine learned surrogate model, as opposed to a ROM that is used in Popov et al. (2021)

The Multi-Fidelity EnKF (MF-EnKF) - forecast step



$N_X = 2$



$N_U = 12$

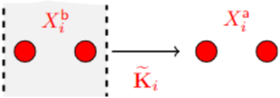
Forecast step

$$X_i^{b,[k]} = \mathcal{M}^X(X_{i-1}^{a,[k]}) + \varepsilon_i^X, \qquad k = 1, \dots N_X$$

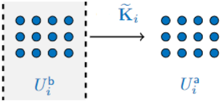
$$U_i^{b,[k]} = \mathcal{M}^U(U_{i-1}^{a,[k]}) + \varepsilon_i^U, \qquad k = 1, \dots N_U$$

- $\mathcal{M}^X$ propagates state forward from time t_{i-1} to t_i
- $\mathcal{M}^U$ is a machine learned surrogate model, as opposed to a ROM that is used in Popov et al. (2021)
- ε_i^X and ε_i^U are the model error terms

The Multi-Fidelity EnKF (MF-EnKF) - Analysis step

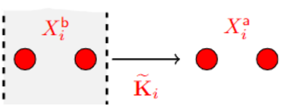


$N_X = 2$

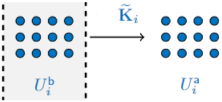


$N_U = 12$

The Multi-Fidelity EnKF (MF-EnKF) - Analysis step



$N_X = 2$



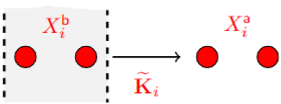
$N_U = 12$

Analysis step

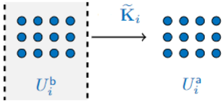
$$X_i^{a,[k]} = X_i^{b,[k]} - \widetilde{K}_i(\mathcal{H}(X_i^{b,[k]}) - y_i - \eta_i), \qquad k = 1, \ldots, N_X$$

$$U_i^{a,[k]} = U_i^{b,[k]} - \widetilde{K}_i(\mathcal{H}(U_i^{b,[k]}) - y_i - \eta_i), \qquad k = 1, \ldots, N_U$$

The Multi-Fidelity EnKF (MF-EnKF) - Analysis step



$N_X = 2$



$N_U = 12$

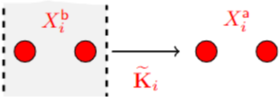
Analysis step

$$X_i^{a,[k]} = X_i^{b,[k]} - \widetilde{K}_i(\mathcal{H}(X_i^{b,[k]}) - y_i - \eta_i), \qquad k = 1, \ldots, N_X$$

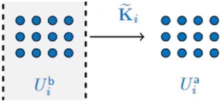
$$U_i^{a,[k]} = U_i^{b,[k]} - \widetilde{K}_i(\mathcal{H}(U_i^{b,[k]}) - y_i - \eta_i), \qquad k = 1, \ldots, N_U$$

- $\tilde{K}_i$ is the Kalman gain matrix, which decides the weighting of the forecast compared to the observation y_i

The Multi-Fidelity EnKF (MF-EnKF) - Analysis step



$N_X = 2$



$N_U = 12$

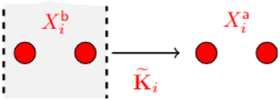
Analysis step

$$X_i^{a,[k]} = X_i^{b,[k]} - \widetilde{K}_i(\mathcal{H}(X_i^{b,[k]}) - y_i - \eta_i), \qquad k = 1, \dots, N_X$$

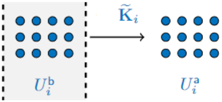
$$U_i^{a,[k]} = U_i^{b,[k]} - \widetilde{K}_i(\mathcal{H}(U_i^{b,[k]}) - y_i - \eta_i), \qquad k = 1, \dots, N_U$$

- $\tilde{K}_i$ is the Kalman gain matrix, which decides the weighting of the forecast compared to the observation y_i
- $\mathcal{H}$ is the observation operator, mapping states to observation space

The Multi-Fidelity EnKF (MF-EnKF) - Analysis step



$N_X = 2$



$N_U = 12$

Analysis step

$$X_i^{a,[k]} = X_i^{b,[k]} - \widetilde{K}_i(\mathcal{H}(X_i^{b,[k]}) - y_i - \eta_i), \qquad k = 1, \dots, N_X$$

$$U_i^{a,[k]} = U_i^{b,[k]} - \widetilde{K}_i(\mathcal{H}(U_i^{b,[k]}) - y_i - \eta_i), \qquad k = 1, \dots, N_U$$

- $\tilde{K}_i$ is the Kalman gain matrix, which decides the weighting of the forecast compared to the observation y_i
- $\mathcal{H}$ is the observation operator, mapping states to observation space
- η_i is the measurement error term

TU Delft

The Multi-Fidelity EnKF (MF-EnKF)

Total analysis

$$Z_i^a = (1 - \lambda)\bar{X}_i^a + \lambda \bar{U}_i^a$$

TUDelft

The Multi-Fidelity EnKF (MF-EnKF)

Total analysis

$$Z_i^a = (1 - \lambda)\bar{X}_i^a + \lambda \bar{U}_i^a$$

How is the information combined?

- Tuning parameter λ

The Multi-Fidelity EnKF (MF-EnKF)

Total analysis

$$Z_i^a = (1-\lambda)\bar{X}_i^a + \lambda \bar{U}_i^a$$

How is the information combined?

- Tuning parameter λ
- Shared Kalman gain matrix $\widetilde{K}_i$

$$\widetilde{K}_i = \mathrm{Cov}(Z_i^b, \mathcal{H}(Z_i^b))(\mathrm{Cov}(\mathcal{H}(Z_i^b), \mathcal{H}(Z_i^b)) + R)^{-1}$$

with $Z_i^b = (1-\lambda)\bar{X}_i^b + \lambda \bar{U}_i^b$ the total background term and R the covariance matrix of measurement errors.

Numerical experiments

We have tested the MF-EnKF with ML surrogate on 2 common toy models:

- Lorenz-96 model
- QG model

Lorenz-96 results

The physical model $\mathcal{M}^X$

- Lorenz-96 equations:

$$\frac{dx^n}{dt} = (x^{n+1} - x^{n-2})x^{n-1} - x^n + F$$

with $n = 0, \ldots, 39$, forcing $F = 8$ and periodic boundary conditions.

TUDelft

Lorenz-96 results

The physical model $\mathcal{M}^X$

- Lorenz-96 equations:

$$\frac{dx^n}{dt} = (x^{n+1} - x^{n-2})x^{n-1} - x^n + F$$

  with $n = 0, \ldots, 39$, forcing $F = 8$ and periodic boundary conditions.

- Initial condition:

$$x_0 = \mathbf{F} + \Delta e_1$$

  with $\mathbf{F}$ the vector with all elements equal to F, which is the steady-state, and $\Delta = 0.01$

Lorenz-96 results

The surrogate model $\mathcal{M}^U$

- Implemented in Tensorflow
- Convolutional Neural Network (CNN)
- 3 convolutional layers, periodic padding
- About 1,000 parameters
- Trained on time-series of length 4,000 from direct simulations of Lorenz-96 model

Lorenz-96 results

Data assimilation setup

- Optimal localization using GC localization function added to EnKF for fairer comparison
- MF-EnKF with $\lambda = 0.5$ as tuning parameter
- Assimilation window: 4000 time steps
- Observations: noisy ($\sigma_{obs} = 1.0$) with 50% of the state observed (only even locations)

Lorenz-96 results

We use a fixed number of 10 full model runs.

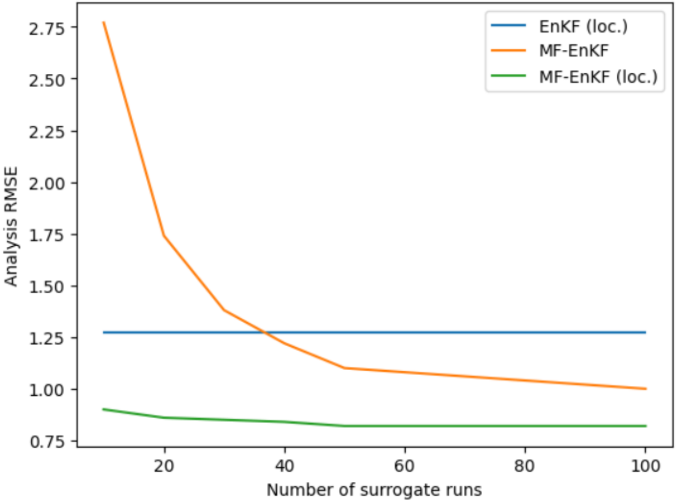


Figure: RMSE for (localized) EnKF and MF-EnKF against N_U

Lorenz-96 results

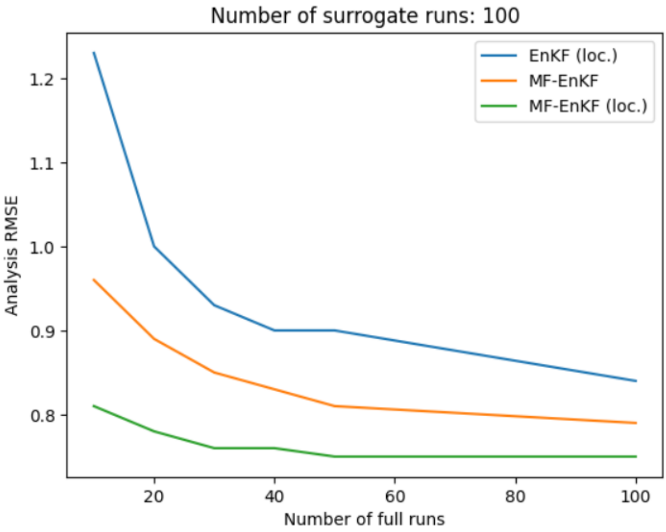


Figure: RMSE for (localized) EnKF and MF-EnKF against N_X

QG model results

The physical model $\mathcal{M}^X$

- 1-layer Barotropic Vorticity equations

$$\partial_t q - \psi_y q_x + \psi_x q_y = 0$$

$$\Delta\psi - \frac{f_0^2}{gH}\psi = q - \beta y$$

- Double gyre setup with wind forcing $F = \frac{L\tau_0}{2\pi\rho_0}\sin(\frac{2\pi y}{L})$
- Free slip boundary conditions
- Initial condition $q_0 = \psi_0 = 0$, spin-up time of 50 years
- Implemented using MQGeometry package from Thiry et al. (2024)

QG model results

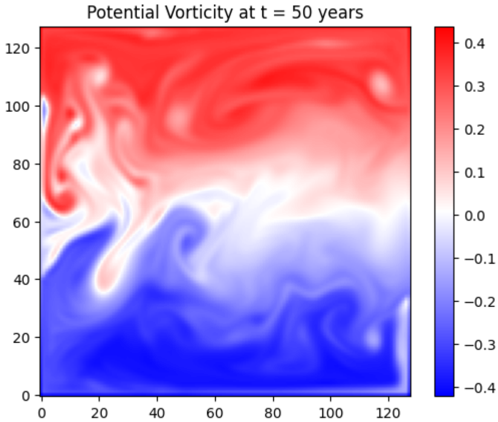


Figure: Initial vorticity (resolution: 128×128) after spin-up

QG model results

The surrogate model $\mathcal{M}^U$

- Implemented in PyTorch
- Convolutional Neural Network (CNN)
- Similar setup as for Lorenz-96 surrogate model
- About 40,000 parameters
- Trained on 1 year of direct simulations from the QG model

QG model results

Data assimilation setup

- MF-EnKF with $\lambda = 0.5$ as tuning parameter
- State vector: streamfunction ψ at every grid point (64×64).
- Localization using GC localization function
- Assimilation window: 1000 time steps
- Observations: noisy ($\sigma_{obs} = 1\%$) observations at 50% of the grid available, randomly selected locations

QG model results - no localization

We use a fixed number of 50 full model runs.

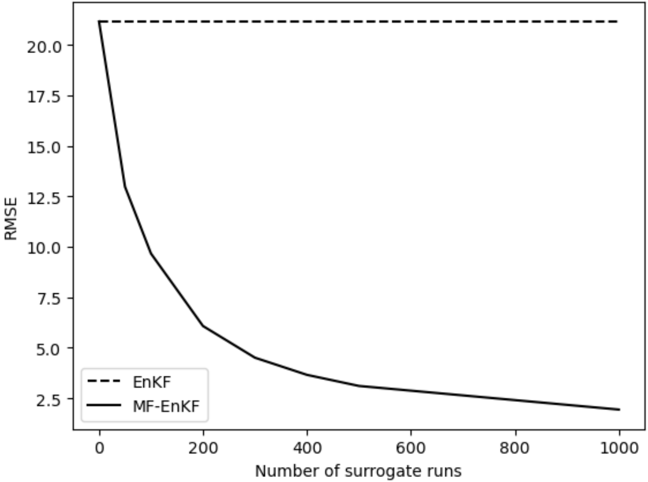


Figure: RMSE for EnKF and MF-EnKF (no localization) against N_U

QG model results - no localization

We use a fixed number of 500 surrogate model runs.

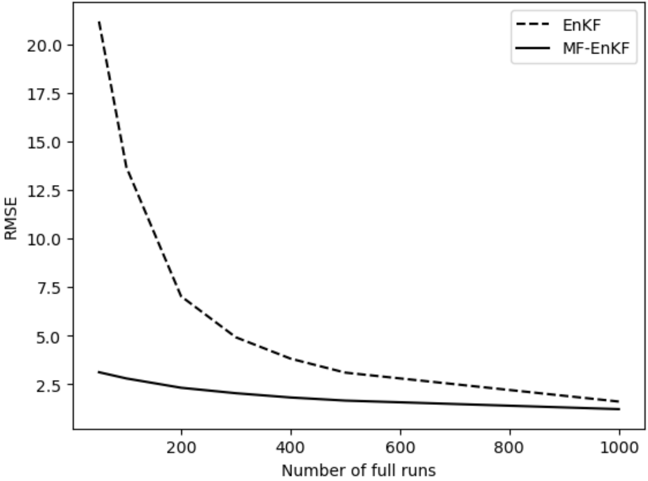


Figure: RMSE for EnKF and MF-EnKF (no localization) against N_X

QG model results - with localization

Fix $N_X = 50$ and $N_U = 500$. Localization radius $r_X = r_U = r$.

r	MF-EnKF	EnKF ($\lambda = 0$)
1	0.89	0.81
2	1.04	1.01
3	1.02	1.13
4	1.06	1.36
5	1.12	1.65
10	1.49	3.57
20	2.44	7.45
64	4.94	15.1

QG model results - with localization

Fix $N_X = 50$ and $N_U = 500$. Localization radius $r_X = r_U = r$.

r	MF-EnKF	EnKF ($\lambda = 0$)
1	0.89	0.81
2	1.04	1.01
3	1.02	1.13
4	1.06	1.36
5	1.12	1.65
10	1.49	3.57
20	2.44	7.45
64	4.94	15.1

Very strict localization needed?

QG model results - with localization

Fix $N_X = 50$ and $N_U = 500$. Localization radius $r_X = r_U = r$.

r	MF-EnKF	EnKF ($\lambda = 0$)
1	0.89	0.81
2	1.04	1.01
3	1.02	1.13
4	1.06	1.36
5	1.12	1.65
10	1.49	3.57
20	2.44	7.45
64	4.94	15.1

Very strict localization needed?

EnKF outperforms MF-EnKF with strict localization → need better surrogate?

Conclusions

1. MF-EnKF with ML surrogate outperforms EnKF for the same number of full model runs
2. MF-EnKF with ML surrogate outperforms localized EnKF for the same number of full model runs **given enough surrogate runs**
3. MF-EnKF can reach similar or improved accuracy with fewer full model runs

Future directions

- Use more realistic assumptions in QG model (finer spatial dimension, more general geometries, less observations)
- Compare influence of different surrogate models (lower-dimensional model VS Neural Network VS linear regression)
- Find optimal value for tuning parameter λ

Thank you for your attention

References

[1] Andrey A. Popov, Changhong Mou, Traian Iliescu, and Adrian Sandu (2021) *A Multifidelity Ensemble Kalman Filter with Reduced Order Control Variates*, SIAM Journal of Scientific Computing, Vol. 43, No. 2, pp. A1134-A1162, 2021.

[2] Thiry et al. (2024) *MQGeometry-1.0: a multi-layer quasi-geostrophic solver on non-rectangular geometries*, Geoscientific Model Development, Vol. 17, No. 4, pp. 1749-1764, 2024.

Extra slides

Influence of λ parameter

$N_X = 10,\ N_U = 100.$

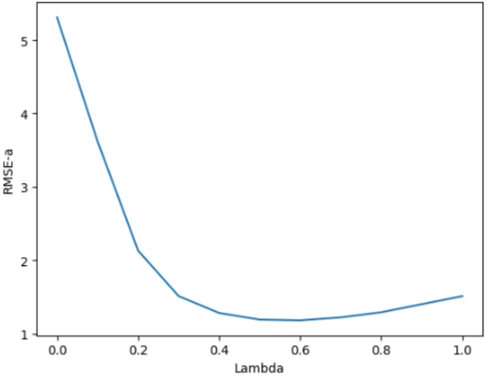


Figure: RMSE for different λ values for Lorenz-96 model (no localization)

$\lambda = 0$: fully trust ensemble of full model runs
$\lambda = 1$: fully trust ensemble of surrogate model runs

Similar result when localization is included.

Influence of surrogate model

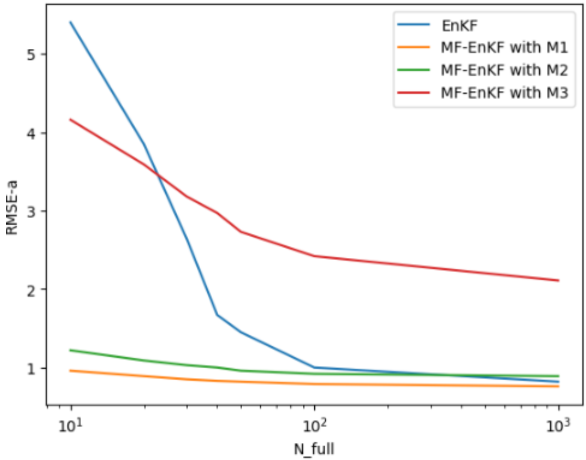


Figure: RMSE for different surrogates for Lorenz-96 with fixed number of surrogate runs (no localization)

$M_1, \ldots, M_3$ are increasingly bad surrogates.

Influence of observed percentage

Fix $N_X = 50$ and $N_U = 500$. Choose $r_X = r_U = r$. Assimilation window: $K = 1000$. Observation locations are selected at random at every time step ($\sim$ every 1.1 hour).

r	RMSE (50%)	RMSE (25%)	RMSE (10%)
1	0.89	2.03	5.35
2	1.04	2.18	5.39
3	1.02	2.10	5.08
4	1.06	2.14	5.12
5	1.12	2.21	5.22
10	1.49	2.80	6.54
20	2.44	4.56	diverges
64	4.94	10.10	diverges

Std. of the streamfunction is about 27.