Lab Talk #2

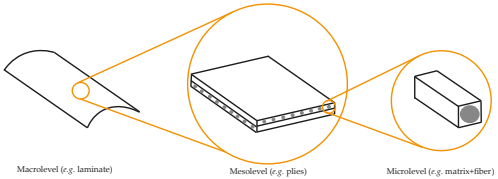
May 23, 2022

Ozgur Tazlan Turan

Expected Loss of Model-Agnostic Meta-Learning (MAML)

- How and Why did I ended up here?
- Learning-to-Learn
- MAML vs Biased Ridge
- Some Results/Conclusions
- What is next?

Constitutive Modeling



Composite Materials

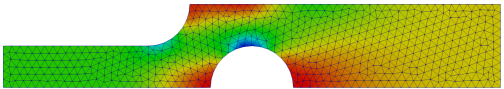
- Why?
- How?

Constitutive Modeling

Composite Materials

- Why?
- How? Experimental $\rightarrow$ time and money

Constitutive Modeling

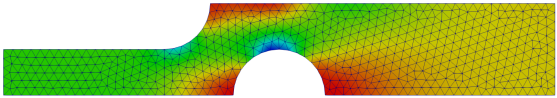


Composite Materials

- Why?
- How? Computational $\rightarrow$ time

Computational Constitutive Modeling (Relationship between force and displacement

Macro-scale Problem

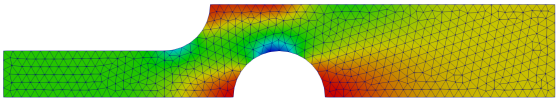


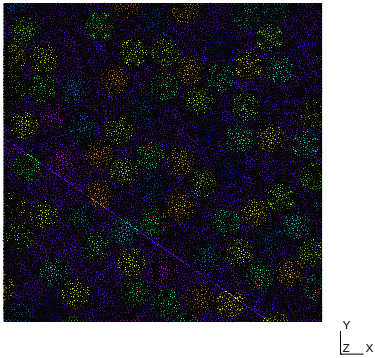
Micro-scale Problem

Computational Constitutive Modeling (Relationship between force and displacement

Micro-scale Problem

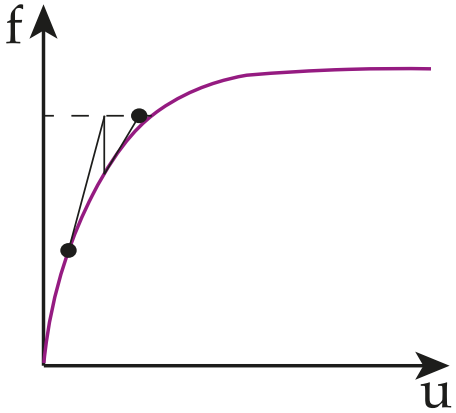
Macro-scale Problem



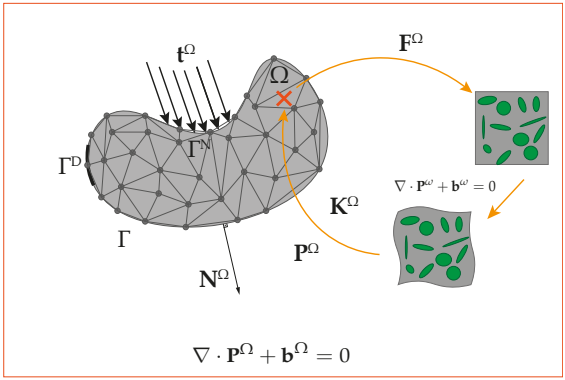


Computational Constitutive Modeling (Relationship between force and displacement

- Inversion of a matrix (dof$\times$dof) at each step (time)

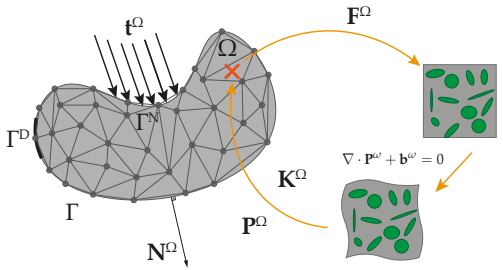


Review

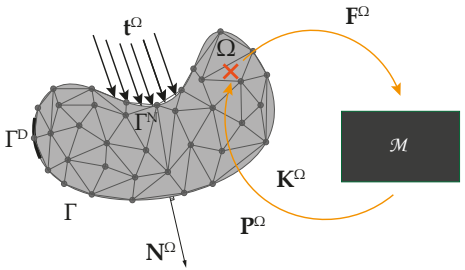


- Computationally model composites
- Time is the bottleneck of the model

Machine Learning to the Rescue



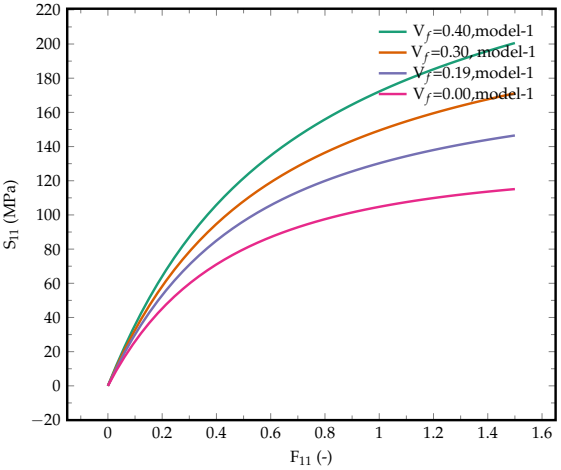
$$\nabla \cdot \mathbf{P}^{\Omega} + \mathbf{b}^{\Omega} = 0$$



$$\nabla \cdot \mathbf{P}^{\Omega} + \mathbf{b}^{\Omega} = 0$$

Machine Learning to the Rescue

- $\mathbf{P}^{\Omega} = \mathcal{C}(\mathbf{F}^{\Omega}, \{c\}_{i=1}^{t})$
- Current Appplication: Focus on one curve to create a surrogate
- Why not consider other curves that might come from some other source!



Learning-to-Learn Intro

Learning

- Task $\rightarrow f : \mathbf{x} \mapsto y$
- Training experience $\rightarrow \mathcal{Z} = \{\mathbf{x}_i, y_i\}_{i=0}^{N}$
- Error measure $\rightarrow \mathcal{L} := \sum_j^M (\mathcal{M}_j - y_j)^2$

Learning-to-learn

- Family of Tasks $\rightarrow \{f_k : \mathbf{x} \mapsto y\}_{k=1}^{K}$
- Training experience for $f_k \rightarrow \mathcal{Z}_k$
- Error measure for each task $\rightarrow \mathcal{L}_k$

- Learning a function vs learning a functional (space of functions!)

Learning-to-Learn Intro

Learning

- Task $\rightarrow f : \mathbf{x} \mapsto y$
- Training experience $\rightarrow \mathcal{Z} = \{\mathbf{x}_i, y_i\}_{i=0}^{N}$
- Error measure $\rightarrow \mathcal{L} := \sum_j^M (\mathcal{M}_j - y_j)^2$

Learning-to-learn

- Family of Tasks $\rightarrow \{f_k : \mathbf{x} \mapsto y\}_{k=1}^{K}$
- Training experience for $f_k \rightarrow \mathcal{Z}_k$
- Error measure for each task $\rightarrow \mathcal{L}_k$

- Learning a function vs learning a functional (space of functions!)

Learning-to-Learn Intro

Learning

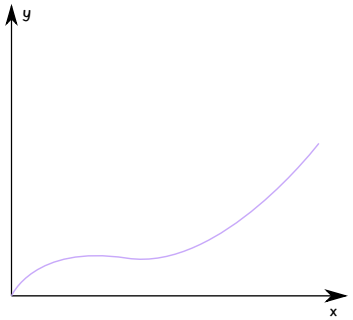
- Task $\rightarrow f : \mathbf{x} \mapsto y$
- Training experience $\rightarrow \mathcal{Z} = \{\mathbf{x}_i, y_i\}_{i=0}^{N}$
- Error measure $\rightarrow \mathcal{L} := \sum_j^M (\mathcal{M}_j - y_j)^2$

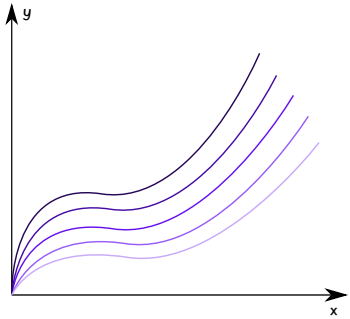
Learning-to-learn

- Family of Tasks $\rightarrow \{f_k : \mathbf{x} \mapsto y\}_{k=1}^{K}$
- Training experience for $f_k \rightarrow \mathcal{Z}_k$
- Error measure for each task $\rightarrow \mathcal{L}_k$

- Learning a function vs learning a functional (space of functions!)

Learning-to-learn Intro





Learning-to-learn

- since 90's
- rooted with LSTM type of meta-learners
- Now, gradient descent based adaptation based NN's $\rightarrow MAML$

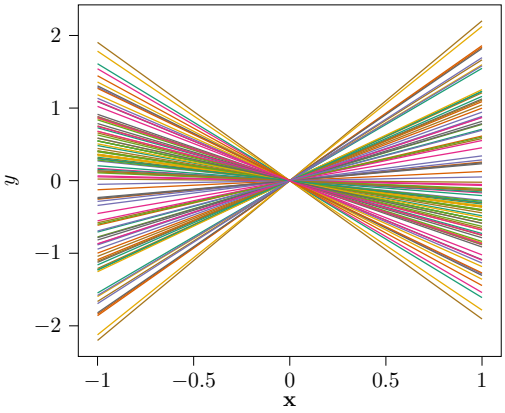
Questions?

- Does the claims for generalization hold?
- Can this framework outperform a single-task learner in expectation?

Problem Setting-1

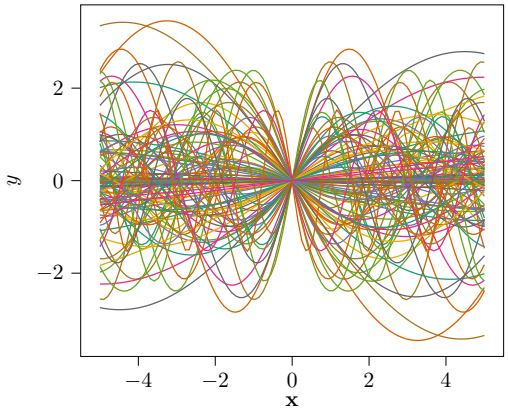
Linear Problem

- $\mathbf{a} \in \mathbb{R}^d \rightarrow p_{\mathcal{T}} \sim \mathcal{N}(m\mathbf{1}, c\mathbf{I})$
- $\mathbf{x} \in \mathbb{R}^d \rightarrow p_{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, k\mathbf{1})$
- $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- $y = \mathbf{a}^{\mathsf{T}}\mathbf{x} + \varepsilon \quad \in \mathbb{R}$
- $\mathcal{Z} := ((x_i, y_i))_{i=1}^N$
- $\hat{\mathcal{M}} \rightarrow$ an estimator trained with $\mathcal{Z}$



Expected Error for an estimator: $\mathcal{E} := \int\int\int (\hat{\mathcal{M}} - y)^2 p(x,y) dx dy p_{\mathcal{Z}} d\mathcal{Z} p_{\mathcal{T}} d\mathcal{T}$

Problem Setting-2



Nonlinear Problem

- $\mathbf{a} \in \mathbb{R}^d \rightarrow p_{\mathbf{a}} \sim \mathcal{N}(\mathbf{1}, c_1\mathbf{I})$
- $\boldsymbol{\phi} \in \mathbb{R}^d \rightarrow p_{\boldsymbol{\phi}} \sim \mathcal{N}(\mathbf{0}, c_2\mathbf{I})$
- $\mathbf{x} \in \mathbb{R}^d \rightarrow p_x \sim \mathcal{N}(\mathbf{0}, k\mathbf{1})$
- $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- $y = \mathbf{a}^{\mathsf{T}}\sin(\mathbf{x} + \phi) + \varepsilon \quad \in \mathbb{R}$
- $\mathcal{Z} := ((x_i, y_i))_{i=1}^N$
- $\hat{\mathcal{M}} \rightarrow$ an estimator trained with N training points

Expected Error for an estimator: $\mathcal{E} := \int\int\int (\hat{\mathcal{M}} - y)^2 p(x, y) dx dy p_{\mathcal{Z}} d\mathcal{Z} p_{\mathcal{T}} d\mathcal{T}$

Models

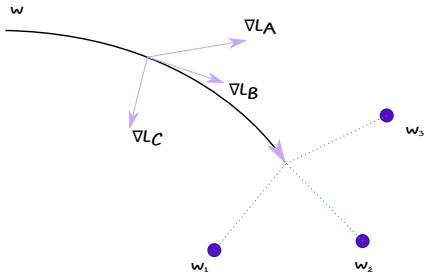
Problem Setting-1

- Linear Model $\rightarrow$ Least-Squares and Gradient Descent
- Ridge $\rightarrow$ with bias and without bias
- MAML

Problem Setting-2

- Kernel Ridge
- MAML

MAML[1]

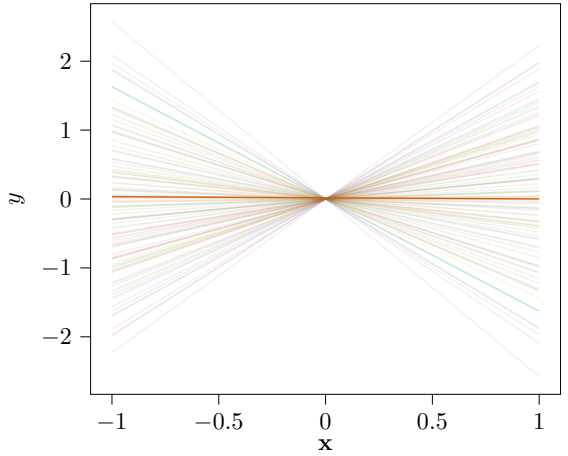


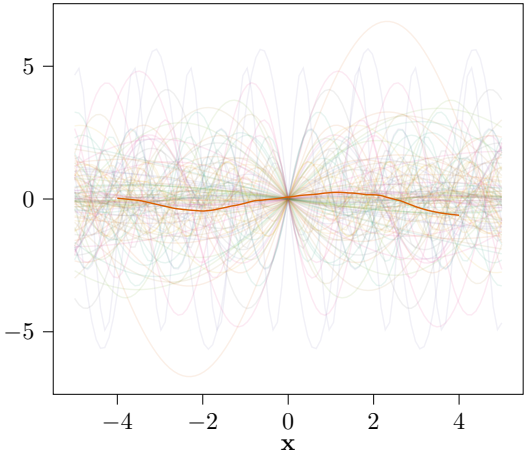
- Sample Tasks
- Sample training experiences from that tasks
- Check the possible loses
- Take average step

For a model $\mathcal{M}$ parametrized by $(\mathbf{w})$

[1] C. Finn, P. Abbeel, and S. Levine (July 2017). "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *arXiv:1703.03400 [cs]*. arXiv: 1703.03400 [cs]

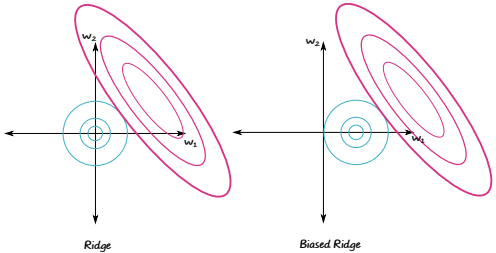
MAML[1]





[1] C. Finn, P. Abbeel, and S. Levine (July 2017). "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *arXiv:1703.03400 [cs]*. arXiv: 1703.03400 [cs]

Biased Ridge[1]



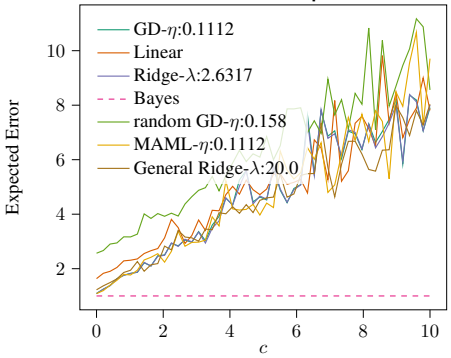
- Sample Tasks
- Sample training experiences from that tasks
- adjust the bias

For a model $\mathcal{M}$ parametrized by $(\mathbf{w})$ minimize $\mathcal{L} + \lambda||\mathbf{w} - \mathbf{h}||_2^2$

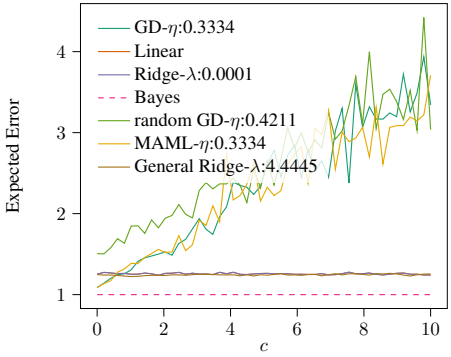
[1] **Denevi2018a**

Problem Setting-1

Limit the number of gradient steps for adaptation to 1 and other parameters regarding the problem is defaulted to 1 as well.



N=1



N=10

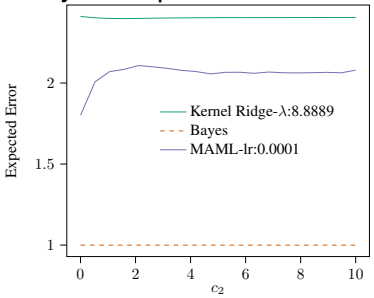
Problem Setting-1

Only consider the $c = [0, 1]$

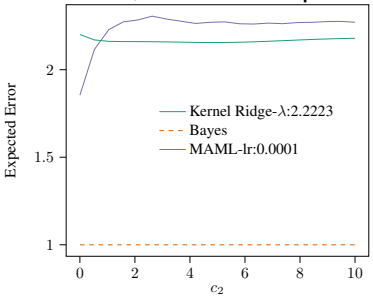
	number of gradient steps									
	1	2	3	4	5	6	7	8	9	10
MAML	1.21	**1.19**	1.20	1.21	1.23	1.24	1.27	1.33	1.46	1.75

Problem Setting-2

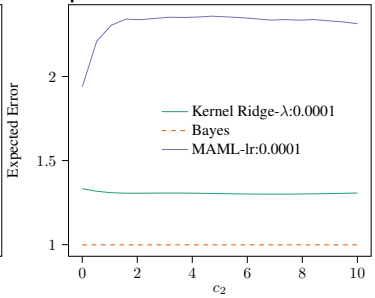
Every other parameter is defaulted to 1, and the adaptation steps used is 5.



N=1



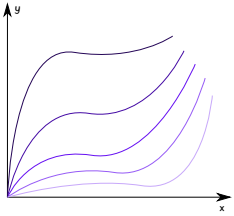
N=10

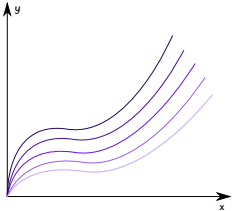


N=50

Conclusions

- Benefit of MAML only for $p_{\mathcal{T}}$ with small variance...
- Number of gradient steps taken is a clear regularizing effect...
- Not, a huge gain though???





Conclusions

- Benefit of MAML only for $p_{\mathcal{T}}$ with small variance...
- Number of gradient steps taken is a clear regularizing effect...
- Not, a huge gain though???

So, the question is...

- Can we come up with a meta kernel ridge model that can outperform MAML?
- I think yes we, can

Remember Kernel Ridge

Thanks to Nonparameteric Representer Theorem for $g \to \mathbb{R}$ be strictly increasing and $\mathcal{L}$ being a loss function that is monotonic, then

$$min_{f \in \mathcal{H}} \quad \mathcal{L} + g(||f||_{\mathcal{H}})$$

has the solution in the form,

$$f(\cdot) = \sum_{i}^{N} \alpha_i k(\cdot, \mathbf{x}_i).$$

$\mathcal{L} = \sum_i^N (f(\mathbf{x}_i) - y_i)^2$. For $g(||f||_{\mathcal{H}}) := \lambda ||f||^2$, the optimal solution is given by:

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}$$

Semiparametric Kernel Ridge

In the Semiparameteric Representer Theorem $\tilde{f} := f + h$ where $h \in span\{\psi_p\}$ where $\{\psi_p\}_{p=1}^M$ are real valued functions and $\mathcal{L} = \sum_i^N (\tilde{f}(\mathbf{x}_i) - y_i)^2$. Then the solution to

$$min_{f \in \mathcal{H}} \quad \mathcal{L} + g(||f||_{\mathcal{H}})$$

has the form

$$f(\cdot) = \sum_i^N \alpha_i k(\cdot, \mathbf{x}_i) + \sum_j^M \beta_j \psi_j(\cdot).$$

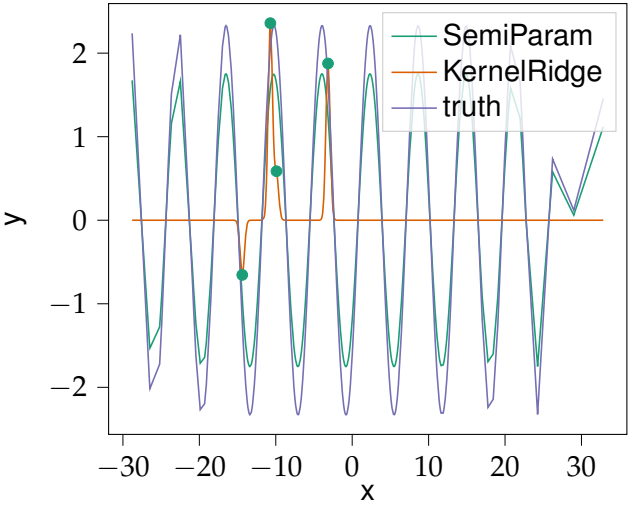
For $g(||f||_{\mathcal{H}}) := \lambda ||f||^2$, the optimal solution is given by:

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1}(\mathbf{y} - \boldsymbol{\psi}\boldsymbol{\beta})$$

$$\boldsymbol{\beta} = \boldsymbol{\psi}^{-1}(\mathbf{y} - \mathbf{K}\boldsymbol{\alpha})$$

Simple Example Case

- Consider the Problem Setting-2
- Assume 10 ground-truth tasks are provided as for $D = 1$
  $\psi(x) = \mathbf{a}^{\mathsf{T}} \sin(\mathbf{x} + \phi)$



- Experiment more with Semi-Parametric Kernel Ridge Method
- Other estimators for ψ? Importance of data for those intermediate models?
- Derivation of Biased Kernel Ridge Method with $g(||f||) = \lambda||f - f_0||^2$
- Look at the Learning Curves and apply to mechanics data!

Thanks!

Algorithm 2 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
6: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)
7: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
8: Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
9: **end for**
10: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 2 or 3
11: **end while**