第 32 回数値流体力学シンポジウム
B05-1

畳み込みニューラルネットワークを用いた非圧縮CFDにおけるポアソンソルバーの高速化

Acceleration of the Poisson solver in incompressible CFD using convolutional neural network

○ 鈴木隆洸, 東大院, 千葉県柏市柏の葉 5-1-5, E-mail:t.suzuki@daedalus.k.u-tokyo.ac.jp
大道勇哉, JAXA, 東京都調布市深大寺東町 7-44-1, E-mail:ohmichi.yuya@jaxa.jp
金森正史, JAXA, 東京都調布市深大寺東町 7-44-1, E-mail:kanamori.masashi@jaxa.jp
鈴木宏二郎, 東大院, 千葉県柏市柏の葉 5-1-5, E-mail:kjsuzuki@k.u-tokyo.ac.jp
Takahiro Suzuki, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba
Yuya Ohmichi, JAXA, 7-44-1 Jindaijihigashi, Chofu, Tokyo
Masashi Kanamori, JAXA, 7-44-1 Jindaijihigashi, Chofu, Tokyo
Kojiro Suzuki, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba

In this study, we proposed a new method to accelerate the convergence of the Poisson solver for the incompressible CFD analysis, based on the deep learning. This method uses Convolutional Neural Network (CNN) to provide an appropriate initial guess of the pressure field for the iteration process in the Poisson solver. Because CNN is good at estimation of the overall structure of the pressure field, the long-wavelength error, which is known to slowly decrease in an iterative method like Gauss-Seidel method, is reduced beforehand by using CNN. Hence, CNN is expected to significantly reduce the number of iterations required for the convergence in the iterative method. The test calculation of the two-dimensional cavity flow problem demonstrated that the convergence of the present CNN-preconditioned Gauss-Seidel method is three times faster than that of the original one. The number of iterations required for the convergence depends on the estimation accuracy of CNN.

1. 緒言

非圧縮性流体の CFD ではポアソン方程式を解くことにより圧力を求める．ポアソン方程式の求解は反復計算を要することから計算負荷が高く，大規模な問題では計算時間の大部分を占めることが多い．よって，CFD を高速化するためにもポアソンソルバーの高速化は重要な課題である．従来の手法としてはマルチグリッド法 (1) などが多く用いられているが，ここではビッグデータ解析の観点から新しい高速化の手法について考える．非圧縮 CFD において連続の式を満たす解が得られた場合，速度場に対応する圧力場は一意に決定される．これは速度場のデータに圧力場の情報が含まれていることを意味し，大量の CFD データを解析することにより速度と圧力との対応関係を獲得できると考えられる．

本研究では，多層のニューラルネットワークを用いてビッグデータの中から規則性を学習する手法であるディープラーニングに注目する．ディープラーニングは画像解析や音声解析，自然言語処理などの分野で一般的に用いられているが，応用範囲は広く方程式のソルバーに対しても使われている (2, 3)．また CFD の高速化については，物体の幾何形状を用いた物体まわりの定常的な速度場の推定 (4) などが行われている．そこで，様々な目的で計算された大量の CFD データを再利用し，ディープラーニングによって速度と圧力との関係を学習することができれば，ポアソンソルバーの高速化が期待できる．

筆者らの先行研究では，多次元データの解析を得意とする畳み込みニューラルネットワーク (Convolutional Neural Network: CNN)(5) を用いることで，速度や速度勾配の情報から圧力の全体的な構造を陽的に推定できることを明らかにした (6)．CNN はニューラルネットワークによる推定値と正解との誤差を最小とするようにネットワークの最適化を行う．ポアソン方程式の解を正解とすることで，CNN は流れ場全体においてポアソン方程式の解との誤差を直接減少させることができる．ゆえに CNN は圧力の全体的な構造の推定，言い換えるとポアソン方程式における長波長の誤差の低減を得意としており，短波長の誤差から収束するガウス-ザイデル法などの反復解法とは反対の特徴を持っている．よって，CNN による圧力の推定を前処理として従来の反復解法と相補的に用いることでポアソン方程式の収束性を高めることができると考えられる．

本研究では，畳み込みニューラルネットワークを前処理としたポアソンソルバーの高速化手法について提案し，キャビティ流れの 2 次元非圧縮 CFD において有効性を検証する．また，圧力の推定精度の異なる CNN モデルを用いることにより，推定精度と高速化の関連について調査する．

2. 手法

2.1 全体の構成

フラクショナルステップ法による非圧縮 CFD では，速度と圧力は以下のように更新される．式 (2) のポアソン方程式の求解を高速化するため，本研究では CNN を使用した．

$$\boldsymbol{V}^{*} = \boldsymbol{V}^{n} + \Delta t\left[-\left(\boldsymbol{V}\cdot\nabla\right)\boldsymbol{V} + \frac{1}{\mathrm{Re}}\nabla^{2}\boldsymbol{V}\right] \tag{1}$$

$$\nabla^{2}p^{n+1} = \frac{1}{\Delta t}\nabla\cdot\boldsymbol{V}^{*} \tag{2}$$

$$\boldsymbol{V}^{n+1} = \boldsymbol{V}^{*} - \Delta t\nabla p^{n+1} \tag{3}$$

本手法のフローチャートを Fig. 1 に示す．黒線で表される従来の解法に対し，赤線で表される CNN の処理が追加されている．CNN はポアソン方程式の右辺項（以下，RHS と呼ぶ）から圧力を陽的に推定するために用いられ，CNN による圧力の推定値を初期値として，ガウス-ザイデ

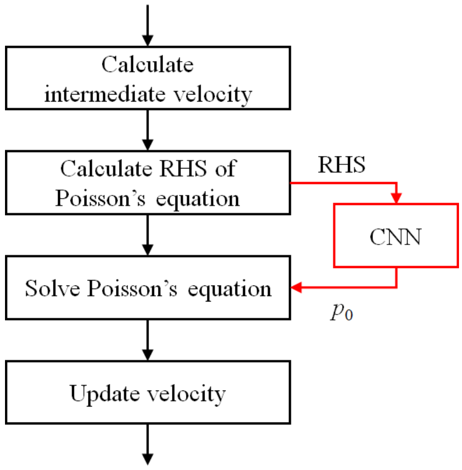


Fig. 1: Flowchart of the proposed method

ル法などの反復解法によりポアソン方程式を解く．CNN はポアソン方程式を解く前に一度だけ呼び出され，反復解法のアルゴリズムを変更する必要はない．よって，既存のコードへの移植が容易であり，シンプルな手順により高速化が可能となる．CNN 及び CNN を用いたポアソンソルバーの高速化の手法について以下に説明する．

2.2　畳み込みニューラルネットワーク

CNN は畳み込み層，プーリング層などにより構成される多層のニューラルネットワークであり，画像認識などの多次元データを扱う問題に多く用いられている．CNN は教師あり学習の一種であり，ネットワークの学習には入力側（画像）と出力側（ラベルや画像など）の訓練データを必要とする．これは問題と正解に対応し，問題に対する CNN の出力が正解に近づくようにネットワークのパラメータを最適化することで，画像の特徴を自動的に学習する．CNN の学習は，入力データに対して入力層から出力層へ向けて正解を予測する順伝播，出力データと正解との誤差を小さくするように出力層から入力層へ向けてネットワークのパラメータを更新する逆伝播の 2 段階から成る．

畳み込み層では，画像に対してフィルタを作用させることによって特徴を抽出する．入力画像は畳み込み層を通すことによってフィルタに対応した構造，特徴マップに変換される．各畳み込み層において複数種類のフィルタが使用されており，フィルタの数だけ特徴マップが得られる．チャンネル数 K の特徴マップを $\boldsymbol{x}$，サイズ $H \times W$ のフィルタを $\boldsymbol{w}$，バイアスを b とすると，l 層目での畳み込み処理は式 (4) で表される．ここで，フィルタ $\boldsymbol{w}$ 及びバイアス b はネットワークのパラメータであり，各層においてこれらの値を最適化することによって画像を最も良く表現する特徴を抽出することができる．

$$z^l_{i,j,k'} = \sum_{k=1}^{K}\sum_{p=1}^{H}\sum_{q=1}^{W} w^l_{p,q,k,k'} x^{l-1}_{i+p,j+q,k} + b^l_{k'} \tag{4}$$

また，ネットワークを多層にすることで表現力を高めるために，式 (5) のように畳み込み処理の出力に対して活性化関数と呼ばれる非線形関数を作用させる．CNN などの多層ニューラルネットワークにおいて多く用いられる活性化関数として，ReLU(Rectified Linear Unit)[7, 8] が挙げられる．ReLU は $h(x) = \max(0, x)$ で表される非線形関数であり，多層ニューラルネットワークの逆伝播計算において誤差の情報が発散，消失しにくい特徴がある．

$$x^l_{i,j,k'} = h\left(z^l_{i,j,k'}\right) \tag{5}$$

プーリング層では画像から一部の領域を取り出し，領域内の最大値や平均値を代表値として出力する．これにより特徴マップの解像度を落とし，特徴の微小な位置のずれに対する感度を鈍くすることができる．本研究では最大値プーリングを採用した．

パラメータ $\boldsymbol{w}$，b は誤差逆伝播法によって最適化される．誤差逆伝播法では，CNN の出力と正解となる訓練データとの誤差を表す損失関数 E を最小とするように，確率的勾配降下法などを用いて以下のようにパラメータを更新する．

$$\begin{aligned} \boldsymbol{w} &\leftarrow \boldsymbol{w} - \epsilon\frac{\partial E}{\partial \boldsymbol{w}} \\ b &\leftarrow b - \epsilon\frac{\partial E}{\partial b} \end{aligned} \tag{6}$$

ここで係数 ϵ は学習率である．ϵ を大きな値とすることで学習が速く進む一方で不安定になりやすい．効率的に学習を進めるために学習率を適応的に調節する方法が多く提案されており，本研究では Adam[9] を採用した．

2.3　CNN を用いたポアソンソルバーの高速化

CFD によって得られた流れ場のデータから RHS を入力，圧力の収束解を正解として CNN の学習を行い，学習された CNN モデルを用いて圧力場の推定を行う．CNN の出力は，流れ場全体において正解である圧力の収束解と近い値となる．CNN による収束解の全体的な構造の再現は，つまりポアソン方程式における誤差の長波長成分が小さいことを意味し，従来の反復解法の弱点を補うことが可能となる．よって，誤差の長波長成分を得意とする CNN を前処理とすることで，短波長成分を得意とする従来の反復解法の収束性を高めることができる．

CNN の入力としては速度，速度勾配などが考えられるが，ここでは圧力との関係が時間刻みに対して不変であるポアソン方程式の RHS を用いた．RHS を入力とすることで異なる時間刻みにおける CFD データに対応することができ，多様な種類の流れ場の学習が可能となる．

CNN の学習に用いる訓練データを作成する際，流れ場全体におけるデータを入出力とすると，1 組のデータセットを用意するために 1 ケースの CFD が必要となり，非常に高コストな計算となる．訓練データ作成の問題を解決するために，流れ場から切り出した小区間（パッチ）におけるデータに対するネットワークを構築した．これにより 1 ケースの CFD から複数の訓練データを作成することができ，訓練データを用意するために必要な CFD データの数を大幅に削減することができる [6]．ポアソン方程式において，境界条件は解に対して影響を及ぼす重要な情報であるが，境界条件は圧力を介して速度にフィードバックされる．よって，パッチ内には境界条件の情報が陰に含まれており，パッチ内における RHS の情報のみに基づいて圧力を推定することができる．また，流れ場全体の情報を使用しないことから，大規模並列計算においてノード間の情報通信を行う必要がなくなる．

3. 解析手順

3.1 訓練データの作成

本研究では，解析対象に 2 次元 Regularized Cavity 流れ [10] を用いた．Regularized Cavity 流れでは正方キャビティの上面壁での流速が以下のように与えられ，その他の壁での流速をゼロとしている．

$$u(x) = 16x^2(1-x)^2 \qquad (0 \leq x \leq 1) \tag{7}$$

2 次元非圧縮ナビエ-ストークス方程式の数値解析として，速度と圧力のカップリングにはフラクショナルステップ法を用いた．対流項の差分は 2 次精度コンシステントスキーム [11]，その他の項は 2 次精度中心差分とし，2 次精度アダムス-バッシュフォース法による時間積分を行った．圧力ポアソン方程式の解法には SOR 法を用いた．計算格子は 256×256 点の等間隔直交格子とし，変数の配置にはスタッガード配置を採用した．レイノルズ数を 1000 から 10000 まで，1000 刻みの 10 条件とし，定常解を計算した．

CNN の学習に使用する訓練データとして，256×256 点の流れ場データから切り出した 32×32 点のパッチを用いた．はじめに，流れ場データに対して縦横 8 格子ずつ走査しながらパッチを切り出し，1 枚の流れ場データから 841 枚のパッチを得た．次に，各パッチを回転，反転させてデータ数を 8 倍とした．これはデータ数を増やすだけでなく，ネットワークに回転，反転不変性を学習させるのに有効である．これらの手法をレイノルズ数 10 条件での CFD データに対して行い，最終的な訓練データ数は 67280 となった．

3.2 ネットワークの学習

CNN のネットワーク構造として U-Net [12] を用いた (Fig. 2)．U-Net は物体の輪郭を検出するセグメンテーションなどに用いられ，少ない訓練データにおいても良い性能を示すことが報告されている．フィルタサイズ 3×3 の畳み込みを 2 回行い，ダウン（アップ）サンプリングにより解像度を変えることでマルチスケールな特徴を抽出している．特徴マップのダウンサンプリングには最大値プーリング，アップサンプリングには転置畳み込みを用いた．出力層には負の値を表現できない ReLU を入れていない．

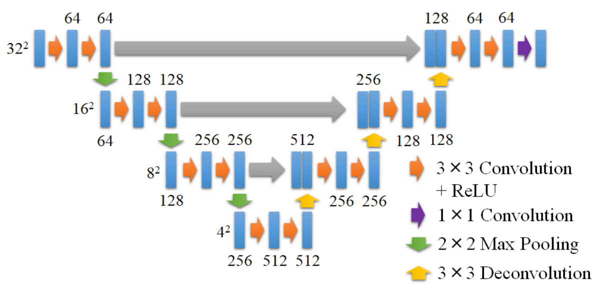


Fig. 2: Network architecture

損失関数は式 (8) のように定義した．ここで，$x^L_{i,j}$ 及び $t_{i,j}$ はそれぞれ各格子点における CNN の出力，正解データを表している．

$$E = \frac{\sum_i \sum_j \left| x^L_{i,j} - t_{i,j} \right|}{\sum_i \sum_j \left| t_{i,j} \right|} \tag{8}$$

各学習ステップにおいて，全訓練データからランダムに抽出された 64 データをミニバッチとしてまとめて学習に用いた．これによりパラメータの更新量のばらつきを抑え，学習を安定化させている．ネットワークの学習には 5×10^6 個のミニバッチを使用した．

CNN モデルはディープラーニング用フレームワーク Tensorflow [13] を用いてコーディングした．GPU には NVIDIA 社製の GeForce® GTX 1080Ti を使用した．

3.3 解析条件

本研究では，CNN を前処理としたポアソンソルバーが高速化に有効であることを検証するために，Re=5000 においてガウス-ザイデル法のみによる計算と CNN を前処理としたガウス-ザイデル法による計算を比較した．また，学習の途中段階での CNN モデルを比較することで，CNN による圧力の推定精度と加速性能との関係を調べた．ここでは，学習途中の 2 ケース及び学習終了時（5×10^6 ミニバッチ）の計 3 ケースにおける CNN モデルを比較した．各ケースに使用されたミニバッチ数を Table 1 に示す．ポアソン方程式の残差を以下のように定義し，各ケースにおいて残差の値が 10^{-4} 以下となるまで反復計算を行った．

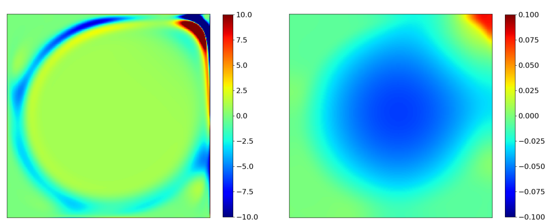
$$R = \frac{1}{N}\sqrt{\sum_i \sum_j \left(\nabla^2 p - \frac{1}{\Delta t} \nabla \cdot \boldsymbol{V}^* \right)^2_{i,j}} \tag{9}$$

Table 1: Number of mini-batches used for the training of each CNN model

	Number of mini-batch
Case1	1×10^5
Case2	1×10^6
Case3	5×10^6

4. 結果・考察

CFD によって得られた Re=5000 における RHS 及び圧力を Fig 3 に示す．RHS を入力，圧力を正解として CNN の学習を行った．学習には GPU を用いて約 42 時間を要した．



(a) RHS (b) Pressure

Fig. 3: RHS of the Poisson's equation and the pressure at Re=5000

CNN の学習が収束したかどうかを確認するために，ミニバッチ数（ステップ数）を横軸，損失を縦軸にとった学習曲線をプロットした．Fig. 4 より学習が進むにつれて損失の値が小さな値に収束していることが分かる．CNN

の出力と正解との誤差は，圧力のノルムに対して 1%以下であった．

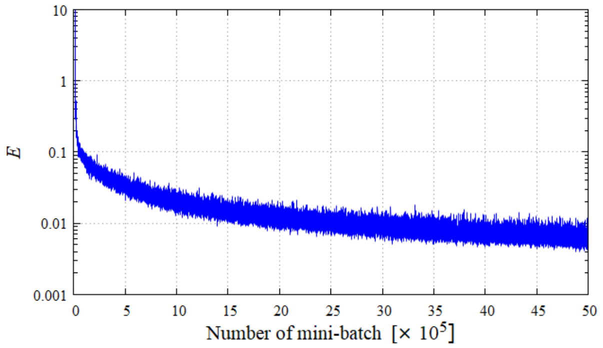
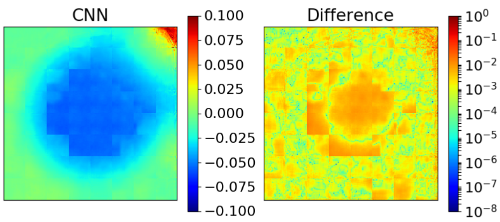
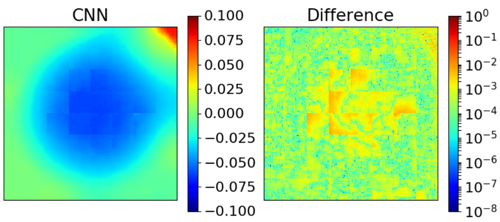


Fig. 4: Learning curve

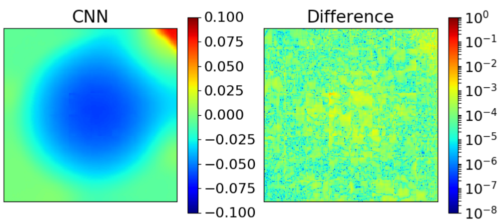
次に，学習が進むにつれて圧力の推定値がどのように変化するかを調べるため，Re=5000 における CNN の出力及び正解との絶対値誤差を可視化した (Fig. 5)．学習曲線が収束していくにつれて，CNN が CFD による圧力を精度良く再現できていることが分かる．学習の初期においては誤差の大きな領域がキャビティの中心や右上の付近に固まって現れているが，学習ステップを重ねることで誤差が小さくなるだけでなく，誤差の大きな領域が分散している．これにより長波長の誤差をより低減できていると見られる．



(a) 1×10^5 mini-batches



(b) 1×10^6 mini-batches



(c) 5×10^6 mini-batches

Fig. 5: CNN outputs at Re=5000

最後に，ガウス-ザイデル法及び CNN を前処理としたガウス-ザイデル法（3 ケース）における収束性を比較した．Re=5000 における，損失と反復回数との比較及び各ケースでの収束性の比較を Table 2，Fig. 6 に示す．CNN を前処理とした方法では反復の初期において残差が急激に減少しており，CNN による収束性の向上が確認され，Case3 ではガウス-ザイデル法と比べて 3.2 倍の加速となった．一方で，反復が進むにつれて残差の減少は鈍化することが分かった．CNN の損失が小さいほど反復回数は少なくなるが，Case2 と Case3 には大きな違いが見られず，収束性の向上は飽和する傾向にある．これには CNN によって除去できなかった，または反復計算の最中に生じた長波長の誤差が要因として挙げられる．前者については，CNN による圧力の推定精度や訓練に用いる正解データの質を向上させる必要がある．後者については CNN にノイズが含まれる以上不可避であるため，圧力の推定値が滑らかな分布になるような CNN の改良などの対策が必要となる．

Table 2: Comparison of estimation accuracy of CNN and number of iterations required for the convergence

	Loss	Iterations
GS	—	73910
CNN+GS (Case1)	8.00×10^{-2}	53475
CNN+GS (Case2)	1.89×10^{-2}	26228
CNN+GS (Case3)	5.22×10^{-3}	22937

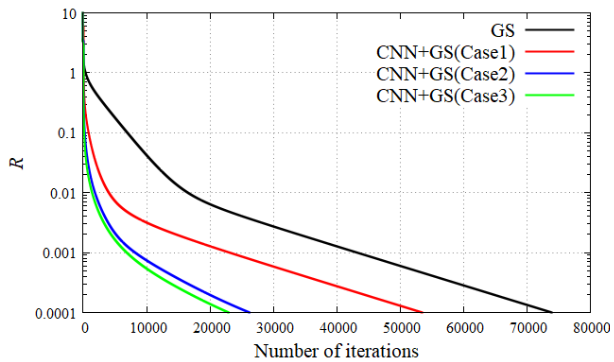


Fig. 6: Comparison of the convergence rates

5. 結言

本研究では，畳み込みニューラルネットワークを前処理としたポアソンソルバーの高速化手法について提案し，フラクショナルステップ法によるキャビティ流れの数値解析においてガウス-ザイデル法の収束性が向上することを確認した．CNN による圧力の推定精度が高いほど反復回数は少なくなるが，推定精度の向上に伴い反復回数が飽和する傾向にあることが分かった．

実際に CFD コードに組み込んで使用するためには，様々な種類の流れに対する CNN の汎用性の検証，一般曲線座標への拡張などが必要であり，今後の検討課題とする．

謝辞

本研究は JAXA 航空技術部門からの受託研究の一環として実施された.

参考文献

(1) Briggs, W. B., Henson, V. E., and McCormick, S. F., "A multigrid tutorial, 2nd edition", SIAM (2000)

(2) Lagaris, I. E., Likas, A., and Fotiadis, D. I., "Artificial neural network for solving ordinary and partial differential equations", IEEE Transactions on Neural Networks, Vol. 9, No. 5 (1998), pp. 987-1000

(3) Lagaris, I. E., Likas, A. C., and Papageorgiou, D. G., "Neural-network methods for boundary value problem with irregular boundaries", IEEE Transactions on Neural Networks, Vol. 11, No. 5 (2000), pp. 1041-1049

(4) Guo, X., Li, W. and Iorio, F., "Convolutional neural network for steady flow approximation", Proceedings of 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016), pp. 481-490

(5) LeCun, Y., Buttou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied document recognition", Proceedings of the IEEE, Vol. 86, Issue 11 (1998), pp. 2278-2324

(6) 鈴木, 鈴木, "CFD 解析へのデータ駆動型アプローチ適用の試み", 第 50 回流体力学講演会/第 36 回航空宇宙数値シミュレーション技術シンポジウム講演集 (2018), 2E07

(7) Nair, V., and Hinton, G. E., "Rectified linear units improve restricted Boltzmann machines", Proceedings of the 27th International Conference on Machine Learning (2010)

(8) Glorot, X., Bordes, A., and Bengio, Y., "Deep sparse rectifier neural networks", Proceedings of 14th International Conference on Artificial Intelligence and Statistics (2011)

(9) Kingma, D. P., and Ba, J. L., "Adam: A method for stochastic optimization", Proceedings of the 3rd International Conference on Learning Representations (2015)

(10) Shen, J., "Hopf bifurcation of the unsteady regularized driven cavity flow", Journal of Computational Physics, Vol. 95, Issue 1 (1991), pp. 228-245

(11) 梶島, "対流項の差分形式とその保存性", 日本機械学会論文集 (B 編), Vol. 60, No. 574 (1994), pp. 186-191

(12) Ronnerberger, O., Fischer, and P., Brox, T., "U-Net: Convolutional networks for biomedical image segmentation", International Conference on Medical Image Computing and Computer-Assisted Intervention (2015)

(13) "TensorFlow", <https://www.tensorflow.org>, (accessed on 10 October, 2018)